



Diagnostic de systèmes complexes par comparaison de listes d'alarmes : application aux systèmes de contrôle du LHC

Nabil Bouchair

► To cite this version:

Nabil Bouchair. Diagnostic de systèmes complexes par comparaison de listes d'alarmes : application aux systèmes de contrôle du LHC. Traitement du signal et de l'image [eess.SP]. Université de Grenoble, 2014. Français. NNT : 2014GRENT025 . tel-01292760

HAL Id: tel-01292760

<https://theses.hal.science/tel-01292760>

Submitted on 23 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Automatique Productique**

Arrêté ministériel : 7 août 2006

Présentée par

Nabil BOUCHAR

Thèse dirigée par « **Sylvie CHARBONNIER** » et

Co-encadrée par « **Philippe GAYET** »

préparée au sein du **Laboratoire GIPSA-lab : Grenoble Images
Paroles Signal Automatique**

dans l'**École Doctorale Electronique Electrotechnique
Automatique Traitement du Signal**

Diagnostic de systèmes complexes par comparaison de listes d'alarmes : Application aux systèmes de contrôle du LHC

Thèse soutenue publiquement le « **16 Avril 2014** »,
devant le jury composé de :

Bernard Riéra

Professeur UFR Sciences Exactes et Naturelles, **CRéSTIC**, Reims
(Rapporteur)

Audine Subias

MCF HDR INSA Toulouse, LAAS-CNRS, Laboratoire d'Analyse et
d'Architecture des Systèmes, Toulouse (Rapporteur)

Christophe Bérenguer

Professeur Grenoble-INP, Gipsa-lab, Grenoble (Membre)

Bernard Dubuisson

Professeur émérite, Université de Technologie de Compiègne
UMR C.N.R.S Heudiasyc, Compiègne (Membre)

Sylvie Charbonnier

MCF HDR Université Joseph Fourier Grenoble, Gipsa-lab, Grenoble,
(Directeur de thèse)

Philippe Gayet

Docteur CERN, Genève (Co-encadrant)



Remerciements

Il est dans un manuscrit de thèse des parties plus complexes que d'autres à rédiger, la partie remerciements est sans nul doute celle que je rédige avec le plus de plaisir. Je veux ici remercier les personnes qui ont compté durant cette thèse qui s'est déroulée au CERN entre la Suisse et la France.

Mes tous premiers remerciements vont naturellement à Sylvie Charbonnier, ma directrice de thèse, qui m'a permis d'accomplir avec succès ce travail de thèse. Elle a su m'encadrer et m'orienter avec ses grandes qualités humaines et scientifiques, tout en me laissant une grande liberté dans mon travail. Je tiens également à la remercier pour sa disponibilité et son soutien sans faille. Mes premières pensées vont également vers Philippe Gayet mon superviseur de thèse au CERN, qui a su me guider et me faire profiter de son expérience. Je le remercie pour la confiance qu'il m'a accordée, sa disponibilité et son intérêt pour le projet.

Je voudrais remercier Audine Subias et Bernad Riera d'avoir accepté d'être les rapporteurs de cette thèse ainsi que Christophe Bérenguer et Bernard Dubuisson d'avoir accepté d'être les examinateurs. Je les remercie également pour les idées qu'ils ont suggérées ainsi que pour leurs remarques pertinentes qui m'ont permis d'améliorer ce manuscrit.

Je souhaiterais également remercier Géraldine Thomas qui a été ma première encadrante au CERN, et qui a grandement œuvré pour que ce qui était un stage au départ devienne une belle aventure de trois ans. Merci également à Renaud Barillère qui m'a encadré au cours de ma première année de thèse et m'a accueilli au sein de sa section ainsi qu'à Hervé Milcent d'avoir pris le relai de ma supervision au sein de la section. Je remercie aussi Benjamin Bradu pour son aide précieuse et son soutien technique.

Ces trois années se sont déroulées au sein de la section EN-ICE-SIC dans laquelle j'ai rencontré des gens formidables avec qui j'ai passé d'excellents moments, je souhaiterais leur dédier ce paragraphe en souvenir des bons moments passés, des cafés partagés, et des fous rires échangés. Je pense plus particulièrement à Omer et Ricardo avec qui j'ai passé d'excellents « afterwork », ainsi qu'à Géraldine, Ben, Filippo, Marco, Mirjam, Ivan, Bart et Vladimir. J'ai également eu la chance de côtoyer les membres de la section PLC qui m'ont accueilli comme l'un des leurs et les remercie pour l'accueil et les moments échangés au travail et en dehors.

Durant cette thèse j'ai pu m'adonner à l'une de mes passions, le football. J'ai eu la chance de jouer dans différentes équipes et j'ai aussi une pensée pour eux. Notamment les joueurs de « el equipo C » avec qui j'ai remporté le tournoi de foot du cern. Mais aussi les joueurs de l'AGFC et notamment à coach Ahmad. Merci aussi aux joueurs de la mutinerie pour tous ces matchs joués ensemble.

Durant ces trois années j'ai partagé avec grand plaisir mon bureau avec Alexey. Je n'aurai pu rêver de meilleur collègue de bureau et qui est devenu plus qu'un ami maintenant, j'ai également découvert la meilleure voisine de bureau possible, Efi, qui est très vite devenue elle aussi une grande amie. Je les remercie tous deux d'être venus assister à ma soutenance, je sais que ça n'a pas été évident à organiser mais vous y étiez ! On formait une drôle d'équipe.

Une thèse ne peut s'accomplir sans un entourage qui vous permet de vous changer les idées quand cela est nécessaire et qui vous soutient dans les moments les plus difficiles. Et j'ai eu au cours de ma thèse l'entourage idéal pour réussir. Tout d'abord je pense aux « membres de la coloc », Mael, Maxence, Chris puis François et Ahmad mais aussi à Adri et Mehdi, avec qui j'ai partagé bien plus qu'un appartement, merci pour toutes les sorties, les voyages, les délires et les fous rires, Les gars vous êtes des machines. Merci aussi à mes amis qui me sont chers et que j'ai pu rencontrer durant ma thèse et avec qui j'ai partagé des moments géniaux, merci à Théodora, Marie, Silvia, Marina, Caroline, Ana, Maria, Mélanie, Charlotte, Matthieu, Tiago, Borja, Daniel. Merci à Cécile pour son soutien précieux en fin de thèse et ces moments passés. J'ai aussi une pensée à l'équipe d'ancien E3, je pense à Nesrine qui vit les mêmes expériences que moi avec sa thèse, je pense aussi aux Momo et à Sofien avec qui j'ai eu plaisir à voyager. Une grosse pensée à André toujours présent pour me faire rire.

Je veux clôturer ce chapitre de remerciements par un paragraphe dédié à ma famille. J'en profite pour vous remercier ici pour tous ce que vous avez fait pour moi et votre soutien même si je ne vous l'ai peut-être pas (assez) dit, je le fais ici : merci Papa, merci Maman, merci Rym et merci Lilia. J'ai également une pensée pour ma famille en Algérie qui sont fiers de moi, je vous embrasse.

Il n'existe pas de forteresse imprenable. Il n'y a que des attaques mal menées.

Vauban

A mes grands-parents qui de là-haut veillent sur moi.

A mes parents et mes sœurs que j'aime.

Résumé

Au CERN (Organisation européenne pour la recherche nucléaire), le contrôle et la supervision du plus grand accélérateur du monde, le LHC (Large Hadron Collider), sont basés sur des solutions industrielles (SCADA). Le LHC est composé de sous-systèmes disposant d'un grand nombre de capteurs et d'actionneurs qui rendent la surveillance de ces équipements un véritable défi pour les opérateurs. Même avec les solutions SCADA actuelles, l'occurrence d'un défaut déclenche de véritables avalanches d'alarmes, rendant le diagnostic de ces systèmes très difficile. Cette thèse propose une méthodologie d'aide au diagnostic à partir de données historiques du système. Les signatures des défauts déjà rencontrés et représentés par les listes d'alarmes qu'ils ont déclenchés sont comparées à la liste d'alarmes du défaut à diagnostiquer. Deux approches sont considérées. Dans la première, l'ordre d'apparition des alarmes n'est pas pris en compte et les listes d'alarmes sont représentées par un vecteur binaire. La comparaison se fait à l'aide d'une distance pondérée. Le poids de chaque alarme est évalué en fonction de son aptitude à caractériser chaque défaut. La seconde approche prend en compte l'ordre d'apparition des alarmes, les listes d'alarmes sont alors représentées sous forme de séquences symboliques. La comparaison entre ces deux séquences se fait à l'aide d'un algorithme dérivé de l'algorithme de Needleman et Wunsch utilisé dans le domaine de la Bio-Informatique. Les deux approches sont testées sur des données artificielles ainsi que sur des données extraites d'un simulateur très réaliste d'un des systèmes du LHC et montrent de bons résultats.

Abstract

In the context of the CERN Large Hadron Collider (LHC), a large number of control systems have been built based on industrial control and SCADA solutions. Beyond the complexity of these systems, a large number of sensors and actuators are controlled which make the monitoring and diagnostic of these equipment a continuous and real challenge for human operators. Even with the existing SCADA monitoring tools, critical situations prompt alarms avalanches in the supervision that makes diagnostic more difficult. This thesis proposes a decision support methodology based on the use of historical data. Past faults signatures represented by alarm lists are compared with the alarm list of the fault to diagnose using pattern matching methods. Two approaches are considered. In the first one, the order of appearance is not taken into account, the alarm lists are then represented by a binary vector and compared to each other thanks to an original weighted distance. Every alarm is weighted according to its ability to represent correctly every past faults. The second approach takes into account the alarms order and uses a symbolic sequence to represent the faults. The comparison between the sequences is then made by an adapted version of the Needleman and Wunsch algorithm widely used in Bio-Informatic. The two methods are tested on artificial data and on simulated data extracted from a very realistic simulator of one of the CERN system. Both methods show good results.

TABLE DES MATIÈRES

0. INTRODUCTION	11
1. CONTEXTE ET PROBLÉMATIQUE	17
1.1. LE LHC ET SON SYSTÈME DE CONTRÔLE	18
1.1.1. <i>Présentation du LHC</i>	18
1.1.2. <i>Le système de contrôle du LHC</i>	20
1.2. OBJECTIFS DE LA THÈSE	24
1.2.1. <i>Difficultés de diagnostic des systèmes complexes</i>	24
1.3. FORMALISATION DU PROBLÈME	25
1.3.1. <i>Cahier des charges</i>	25
1.3.2. <i>Approche adoptée</i> :.....	26
2. ÉTAT DE L'ART	28
2.1. DÉFINITION DU VOCABULAIRE PROPRE AU DIAGNOSTIC.	29
2.2. ÉTAT DE L'ART DES MÉTHODES DE DIAGNOSTIC À L'AIDE D'INFORMATIONS BINAIRES.....	30
2.3. MÉTHODES DE DIAGNOSTIC À BASE DE CONNAISSANCES.....	30
2.3.1. <i>Les systèmes expert</i>	30
2.3.2. <i>Diagnostic à base de modèles</i>	31
2.4. MÉTHODES DE DIAGNOSTIC À BASE DE DONNÉES	33
2.4.1. <i>Corrélation d'alarmes</i>	33
2.4.2. <i>Diagnostic de défauts</i>	34
2.5. POSITIONNEMENT.....	37
3. SIMULATEUR DES SYSTÈMES GAZ.....	38
3.1. ÉTUDE DE CAS : LE SYSTÈME GAZ.....	39
3.1.1. <i>Chambre Multifilaire Proportionnelle</i>	39
3.1.2. <i>Le « système gaz »</i>	40
3.2. MODÉLISATION DU SYSTÈME	43
3.2.1. <i>Ecosimpro</i>	43
3.2.2. <i>Création d'une bibliothèque de composants du système Gaz</i>	44
3.2.3. <i>Construction du modèle complet par liaison des composants</i>	46
3.2.4. <i>Intégration au système de contrôle</i>	47
3.2.5. <i>Simulation des défauts et extraction des listes d'alarmes</i>	49
3.3. VALIDATION DU SIMULATEUR	50
3.3.1. <i>Comparaison des variables physiques</i>	50
3.3.2. <i>Comparaison des listes d'alarmes</i>	52
3.3.3. <i>Avis des experts</i>	52
3.4. CONCLUSION	53
4. DIAGNOSTIC PAR COMPARAISON DE VECTEURS D'ALARMES.....	54
4.1. DÉFINITION DU PROBLÈME/FORMALISME	55
4.2. PROTOTYPES.....	56
4.2.1. <i>Définition</i>	56
4.3. POIDS.....	57
4.3.1. <i>Définition</i>	57
4.3.2. <i>Analyse des poids</i>	60

4.4.	DÉFINITION DE LA DISTANCE PONDÉRÉE <i>Dvp</i>	62
4.5.	MÉTHODES DE DÉCISIONS :	63
4.5.1.	<i>Rejet d'un vecteur d'alarmes</i> :	63
4.5.2.	<i>Affectation d'un vecteur d'alarmes à un défaut contenu dans la base d'apprentissage</i>	66
4.6.	CONCLUSION	71
5.	OUTILS D'ANALYSES DE PERFORMANCES	73
5.1.	VALIDATION DES ALGORITHMES	74
5.1.1.	<i>Base d'apprentissage/ Base de validation</i>	74
5.1.2.	<i>Validation croisée – méthode du « leave-one-out »</i>	75
5.2.	MESURES DE PERFORMANCES	76
5.2.1.	<i>Courbe COR</i>	76
5.2.2.	<i>Matrices de confusion</i>	78
5.3.	CONCLUSION	79
6.	ANALYSE DES PERFORMANCES DES MÉTHODES VECTORIELLES	81
6.1.	DISTANCES BINAIRES.....	82
6.1.1.	<i>La distance de Hamming</i>	82
6.1.2.	<i>La distance définie dans (Charkaoui, et al., 2005).</i>	82
6.2.	ANALYSE DES PERFORMANCES EN UTILISANT UNE BASE DE DONNÉES ARTIFICIELLE	83
6.2.1.	<i>Création d'une base artificielle</i>	83
6.2.2.	<i>Résultats obtenus</i>	86
6.3.	ANALYSE DES PERFORMANCES EN UTILISANT LA BASE DE DONNÉES DU CERN.....	95
6.3.1.	<i>Analyse des alarmes et des poids</i>	95
6.3.2.	<i>Performances en diagnostic</i>	99
6.3.3.	<i>Apport de la distance <i>Dvp</i></i>	106
6.4.	ÉVOLUTION DU DIAGNOSTIC AU COURS DU TEMPS	111
6.5.	CONCLUSION	112
7.	DIAGNOSTIC À L'AIDE DE COMPARAISON DE SÉQUENCES D'ALARMES	114
7.1.	FORMALISME	115
7.1.1.	<i>Présentation des données</i>	115
7.1.2.	<i>Définition d'une séquence d'alarmes</i>	116
7.1.3.	<i>Exemple</i>	116
7.1.4.	<i>Notations</i>	117
7.1.5.	<i>Problématique et algorithmes de recherche de séquences</i>	117
7.2.	SIMILARITÉ ENTRE DEUX SÉQUENCES SYMBOLIQUES – L'ALGORITHME DE NEEDLEMAN AND WUNSCH.....	118
7.2.1.	<i>Alignement de deux séquences</i>	118
7.2.2.	<i>Similarité entre deux séquences symboliques</i>	119
7.2.3.	<i>Algorithme de Needleman et Wunsch - alignement de deux séquences</i>	119
7.3.	EXTRACTION DE PROTOTYPES SÉQUENTIELS	123
7.3.1.	<i>Alignement multiple</i>	123
7.3.2.	<i>Extraction de la séquence minimale</i>	128
7.4.	INFLUENCE DES PARAMÈTRES ω , C ET λ	130
7.4.1.	<i>Paramètre λ</i>	130
7.4.2.	<i>Paramètre ω</i>	130
7.4.3.	<i>Paramètre C</i>	131
7.5.	MÉTHODES DE DÉCISION	131

7.5.1.	Rejet d'une séquence d'alarmes	131
7.5.2.	Affectation d'une séquence d'alarmes à un défaut contenu dans la base d'apprentissage	132
7.6.	CONCLUSION	132
8.	ANALYSE DES PERFORMANCES DE LA MÉTHODE DE DIAGNOSTIC PAR COMPARAISON DE SÉQUENCES	133
8.1.	ANALYSE DE PERFORMANCES SUR UNE BASE DE DONNÉES ARTIFICIELLE.	134
8.1.1.	Création d'une base artificielle	134
8.1.2.	Analyses de performances	138
8.1.3.	Conclusions	151
8.2.	ÉTUDES SUR LA BASE DE DONNÉES DU CERN.....	151
8.2.1.	Influence des paramètres ω et λ	151
8.2.2.	Courbes COR en fonction de λ	151
8.2.3.	Courbes COR en fonction de ω	152
8.2.4.	Comparaison avec la méthode vectorielle, Dvp	153
8.3.	COMPARAISON DES MÉTHODES Dvp ET SimNWnorm	159
8.3.1.	Apport de l'ordre d'apparition des alarmes dans l'aide au diagnostic	160
8.3.2.	Apport de Dvp par rapport à SimNWnorm	163
8.3.3.	Étude des erreurs de diagnostics	164
8.3.4.	Conclusion sur la base de données du CERN	166
8.4.	AIDE À L'OPÉRATEUR	166
8.4.1.	Connaissances extraites de la méthode vectorielle.....	166
8.4.2.	Connaissances extraites des prototypes séquentiels	168
8.4.3.	Représentation séquentielle et aide à l'opérateur.	169
8.5.	CONCLUSION	171
9.	CONCLUSION	173
9.1.	MÉTHODE DE DIAGNOSTIC PAR COMPARAISON DE VECTEURS D'ALARMES.....	174
9.2.	MÉTHODE DE DIAGNOSTIC PAR COMPARAISON DE SÉQUENCES D'ALARMES.....	175
9.3.	BILAN SUR LES DEUX MÉTHODES	176
9.4.	PERSPECTIVES	178
10.	BIBLIOGRAPHIE	179
11.	ANNEXE	186

0. Introduction

Le CERN, Organisation Européenne pour la Recherche Nucléaire, a conçu le Large Hadron Collider (LHC) mis en opération en 2008. Le LHC est un accélérateur de particules permettant de faire circuler deux faisceaux de protons à une vitesse proche de celle de la lumière et de réaliser des collisions dans des chambres instrumentées spécifiques, appelées détecteurs, où les particules générées lors de la collision vont pouvoir être analysées. Le LHC est une formidable machine, composé d'un anneau de 27 km de circonférence, enfoui à 100 mètres sous terre. C'est un équipement extrêmement complexe, composé d'un ensemble de systèmes inter-connectés, équipés de milliers de capteurs, d'actionneurs, de régulateurs, qui assurent le fonctionnement optimal du système. La moindre défaillance en un point des systèmes peut générer un arrêt de la machine et stopper l'acquisition de données physiques pour une longue durée.

Compte tenu de la complexité de la machine et de la nécessité de la maintenir en opération en continu afin de pouvoir réaliser pendant plusieurs mois des collisions de particules et fournir aux physiciens un grand nombre de données, le système est surveillé en permanence par des équipes d'opérateurs ayant en charge différents sous-systèmes du LHC (le système cryogénique chargé de maintenir la température à 1.9K, les détecteurs, ...), le LHC étant bien trop complexe pour être surveillé dans sa globalité par une seule équipe d'opérateurs. Leur mission est de s'assurer du bon fonctionnement du système et de repérer au plus tôt tout défaut pouvant apparaître. Leur tâche consiste à scruter les écrans de contrôle du système de supervision qui, par le biais de synoptiques, vont présenter de manière ergonomique l'état courant du système. L'opérateur chargé de la surveillance a accès aux informations délivrés par les capteurs en temps réel, aux informations sur l'état de différents éléments (marche ou arrêt des actionneurs, ...). Il est alerté, par le biais d'une alarme générée par le système de supervision, dès que la valeur d'une variable mesurée par un capteur dépasse un seuil prédéfini. Le système de supervision détecte donc des situations anormales (variables dont la valeur est en dehors de la plage de fonctionnement normal) mais ne propose aucune analyse automatique de la situation. La tâche de diagnostic, qui consiste à reconnaître l'origine du défaut, est entièrement laissée à l'opérateur.

La plupart du temps, une alarme ne se déclenche pas seule mais c'est plutôt une avalanche d'alarmes qui retentit aux oreilles de l'opérateur, de par les interconnexions des différents éléments. Cette propagation d'alarmes a deux origines, elle peut être liée soit à une propagation physique de la panne (exemple : une température trop haute dans une chambre qui se propage alors sur toutes les autres chambres, déclenchant alors successivement de nombreuses alarmes) ou alors une cause « logique », un événement particulier (une pression trop élevée pouvant mettre en péril l'intégrité physique d'un sous-système par exemple) va alors provoquer l'arrêt ordonné par un automate du dit sous-système déclenchant une succession d'alarmes. Ces deux événements peuvent alors s'enchaîner jusqu'à l'arrêt complet du processus.

La tâche de diagnostic peut donc être extrêmement complexe pour l'opérateur. Elle consiste à retrouver, parmi toutes ces alarmes, celles qui sont pertinentes pour l'analyse car

directement liées au défaut et celles qui ne sont que des conséquences ou du « bruit ». Cette analyse des informations fournies par le système de supervision peut être longue et fastidieuse, ajoutant du stress à celui généré par cette situation anormale. Elle demande une très bonne connaissance du système par l'opérateur, dont la formation peut prendre plusieurs années avant de devenir un expert.

L'objectif de cette thèse est de proposer des solutions pour aider l'opérateur à reconnaître rapidement l'origine du dysfonctionnement. Le but de l'étude est d'utiliser les alarmes délivrées par le système de supervision qui représente l'information principalement analysée par l'opérateur de surveillance lorsqu'un défaut survient. Le but du système d'aide à la décision est de fournir une - ou un ensemble - de suggestions sur l'origine du défaut, à partir de l'ensemble des alarmes binaires générées par le système de supervision, depuis l'apparition de la première alarme et sur un horizon temporel à définir, la durée la plus longue de la période d'analyse étant l'instant où la situation est stabilisée, c'est-à-dire qu'aucune nouvelle alarme n'est plus produite.

Notre objectif est de proposer des méthodes génériques, pouvant s'appliquer sur différents sous-système du LHC. Nous ne nous orienterons donc pas vers des méthodes à base de modèles de connaissance, qu'ils soient représentés par des équations différentielles, des réseaux de Pétri ou des bases de règles fournies par un expert. En effet, les systèmes du LHC sont extrêmement complexes, l'élaboration d'un modèle nécessiterait une phase d'étude très longue. De plus, le modèle d'un système ne serait pas transposable à un autre système. Nous nous orienterons vers des méthodes d'analyse de données, où la connaissance du système est fournie par une base d'exemples précédemment rencontrés par le système et enregistrés dans une base de cas.

Nous nous plaçons donc dans l'hypothèse où nous disposons d'une base de données constituée de plusieurs exemples de défauts pouvant se produire sur le système. Chaque exemple de défaut a été étiqueté par un expert du système. Il se traduit par un ensemble d'alarmes généré par le système de supervision sur l'occurrence de ce défaut. Notre objectif est de proposer des méthodes permettant, sur l'occurrence d'un défaut inconnu, de déterminer si le défaut correspond à un défaut présent dans la base d'exemples et si c'est le cas, de reconnaître le défaut particulier. La problématique est particulièrement complexe dans le contexte du LHC car le nombre d'alarmes pouvant se produire quand un défaut apparaît est très important, compte tenu de la très forte instrumentation des systèmes. De plus, la séquence d'alarmes produite est fortement variable : les mêmes alarmes ne sont pas nécessairement présentes sur différents exemples d'un même défaut et leur ordre d'apparition peut être modifié. Il va donc falloir extraire de la liste d'exemples disponibles la signature spécifique du défaut.

Dans le cadre de cette thèse, nous proposerons deux types de méthodes distinctes, qui diffèrent dans la manière de considérer l'information disponible.

- une première approche consistera à considérer l'ensemble des informations disponibles comme un vecteur composé d'alarmes non ordonnées dans le temps. La seule information prise en compte sera la présence ou l'absence de l'alarme parmi la liste des alarmes produites, indépendamment de son ordre d'apparition.

- une deuxième approche considérera l'information comme une séquence d'alarmes, où l'ordre d'apparition de l'alarme joue un rôle important.

Pour produire les données nécessaires à notre étude et évaluer les performances des méthodes proposées sur des données propres au LHC, nous avons construit, au cours de cette thèse, le simulateur d'un des systèmes du LHC, le système GAZ, chargé de la distribution et de la régulation de la composition du mélange des gaz dans les détecteurs de particules. En effet, nous ne disposions pas en début de thèse de base de données composée de plusieurs exemples de défauts. Nous ne souhaitons pas créer des données à partir d'un système simulé classique ou utiliser des benchmarks déjà existants car nous souhaitons disposer de données représentatives de systèmes du LHC. Notre choix s'est porté vers le système GAZ car c'est un système typique utilisant une architecture de contrôle standard au CERN. De plus, la surveillance de ces systèmes se fait souvent par des opérateurs non experts, qui ne sont que de simples utilisateurs, et qui ont donc besoin de beaucoup plus d'assistance lors d'un défaut du système. Étant donné qu'il n'est pas possible de travailler directement sur le système (en effet il n'est pas possible d'arrêter les accélérateurs pour des expérimentations), il a été décidé de développer un simulateur des systèmes gaz afin d'avoir une liberté d'action totale pour produire des défauts. Cette approche a également été encouragée par le fait que le CERN a conçu une méthodologie permettant de développer assez rapidement un simulateur et de l'intégrer directement au système de contrôle réel. Cela permet de disposer d'un simulateur extrêmement réaliste.

Le manuscrit s'organise de la manière suivante.

Le Chapitre 1

Il est consacré à la présentation du CERN. Les principaux objectifs de l'organisation ainsi que les différentes infrastructures y sont exposées. Le système de supervision et les données utilisées pour le diagnostic (les listes d'alarmes) y sont décrits.

Le Chapitre 2

Dans ce chapitre, après une définition des différents termes associés au monde du diagnostic, on procédera à un état de l'art de la problématique du diagnostic de systèmes industriels, et on fera un tour d'horizon des méthodes de diagnostic à base de données binaires.

Le Chapitre 3

Décrit la méthodologie de développement du simulateur, ainsi que le logiciel utilisé. Il s'agit d'une modélisation orientée objet où chaque composant du système est modélisé indépendamment à l'aide d'équations algèbro-différentielles, puis les composants sont reliés afin de créer le système complet. On présente donc dans le chapitre la bibliothèque de composants utilisés dans le cadre de la thèse ainsi que le système complet issue de la liaison des composants de la bibliothèque. Une partie du chapitre est également consacrée à la validation du simulateur. Enfin, son intégration dans le système de contrôle est présentée.

Le Chapitre 4

Expose la méthodologie de diagnostic reposant sur la comparaison de vecteurs d'alarmes à l'aide d'une distance pondérée. Les méthodes de détermination des poids décrivant l'importance de chaque alarme sont présentées après une mise en place du formalisme, ainsi que les différentes méthodes d'aide à la décision envisagées dans le cadre de ce travail.

Le Chapitre 5

Décrit les différents outils utilisés pour évaluer les performances des méthodes de diagnostics présentées dans ce manuscrit.

Le Chapitre 6

Les performances de la méthode de diagnostic développée au chapitre 4 sont évaluées dans ce chapitre à l'aide des différents outils de diagnostic décrits au chapitre 5. Les performances sont évaluées dans un premier temps sur une base de données créée artificiellement à l'aide de générateurs aléatoires puis sur la base de données extraite du simulateur du système gaz développé et exposé au chapitre 3.

Le Chapitre 7

Expose la méthodologie de diagnostic reposant sur la comparaison de séquences d'alarmes. La comparaison des séquences se fait à l'aide d'algorithmes dérivés de méthodes utilisées en Bio-Informatique. Comme pour la méthode vectorielle, après une mise en place du formalisme, les méthodes de comparaison de séquences sont exposées et les techniques de décision présentées.

Le Chapitre 8

Les performances de la méthode de diagnostic par comparaison de séquences sont évaluées à l'aide des outils développés dans le chapitre 5, dans un premier temps sur différentes bases artificielles, puis dans un second temps, sur la base extraite des listes d'alarmes du CERN. Les performances sont à chaque fois comparées à celles de la méthode par comparaison de vecteurs d'alarmes.

1. Contexte et problématique

1.1. Le LHC et son système de contrôle

1.1.1. Présentation du LHC

1.1.1.1. CERN

Le CERN, Conseil Européen pour la Recherche Nucléaire devenu par la suite l'Organisation Européenne pour la Recherche Nucléaire, a été créée en 1954 par une convention entre 12 états européens. Il compte aujourd'hui 20 états membres et 8 états observateurs. L'objectif de cette convention était de mettre en commun les moyens humains, techniques et financiers de chaque pays afin de concevoir et d'utiliser de grands instruments pour la physique nucléaire et la physique des particules.

Les recherches en physique des particules consistent en l'étude des constituants élémentaires de la matière, leurs rayonnements, ainsi que leurs interactions. Pour ce faire, les physiciens utilisent des accélérateurs dans lesquelles des faisceaux de particules (protons, électrons...) sont accélérées à des vitesses proches de celle de la lumière et mises en collision. Deux paramètres principaux caractérisent l'efficacité d'un accélérateur : l'énergie produite lors de la collision et la luminosité (qui exprime la concentration en protons sur la section de croisement des deux faisceaux), ce qui permet d'évaluer la probabilité d'avoir des collisions à l'intersection des deux faisceaux.

1.1.1.2. LHC

Avec ses 27 kilomètres de circonférence, le LHC est le plus grand accélérateur de particules du monde. Il est enterré à 100m sous terre au niveau de la frontière franco-suisse. Deux faisceaux de protons circulent à une vitesse de 299 997 km/s, soit 0.99999 fois la vitesse de la lumière dans ce gigantesque anneau où règne le vide. L'énergie de collision de 14 Tev et sa luminosité 10 fois supérieures à tout autre accélérateur jamais créé en font l'accélérateur le plus puissant et le plus perfectionné du monde. Pour courber les trajectoires et focaliser les faisceaux de particules d'une telle énergie, il faut générer un champ magnétique de plusieurs teslas. Pour cela, le LHC est doté de 4000 aimants, disposés le long de l'accélérateur. Pour produire ce champ magnétique, les aimants doivent être traversés par un courant électrique d'une intensité de l'ordre de 15 000A, ce qui ne peut être réalisé que par des aimants supra-conducteurs. La supra-conductivité permet de faire circuler des courants de fortes intensités car la résistance électrique du matériau supra conducteur est quasiment nulle. Les bobines des aimants contiennent du niobium, un matériau qui devient supra-conducteur quand sa température est abaissée à 9.2 K. Cependant, pour assurer une parfaite répartition de la température et de meilleures performances, la température des aimants du LHC est réglée à 1.9K, à l'aide d'hélium liquide refroidi par d'immenses unités cryogéniques.

Les collisions entre particules ont lieu en différents points de l'accélérateur, appelés expériences. Ces expériences sont au nombre de 4 dans le LHC : CMS, ATLAS, ALICE,

LHCB. Elles ressemblent à des ‘oignons’ cylindriques, pouvant mesurer jusqu’à 14 m de diamètre et remplies de systèmes électroniques sophistiqués. Les collisions se produisent exactement au centre de l’expérience où les différentes propriétés des particules produites sont mesurées par des détecteurs.

Au plus près du faisceau se trouvent des capteurs permettant d’enregistrer les traces des particules issues de la collision, et de suivre leurs trajectoires. Viennent ensuite les équipements de mesure de l’énergie, les calorimètres, dans lesquels la plupart des particules finissent leur parcours. La couche extérieure de l’oignon se compose elle aussi de trajectographes permettant d’identifier toute particule détectable arrivant aussi loin. Chaque expérience est composée d’un ensemble de systèmes de mesure de particules, tous différents par leur rôle et leur technologie. Par conséquent, une expérience n’est pas uniquement un assemblage de capteurs mais plutôt un ensemble cohérent de systèmes électroniques, informatiques, etc., dédiés à la mise en œuvre de détecteurs et sous-détecteurs.

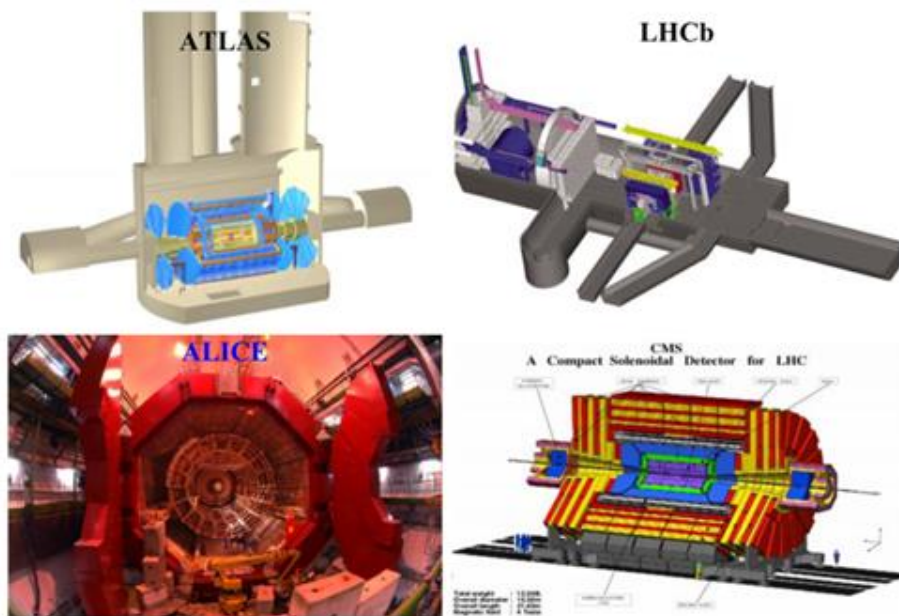


Figure 1-1 Les expériences du LHC

L’immensité des installations ainsi que la multiplicité des différents éléments nécessaires au fonctionnement du LHC en font une machine extraordinairement complexe, où de nombreux systèmes et technologies différents cohabitent. Ces procédés sont contrôlés par des centaines de régulateurs, équipés de centaines de capteurs et d’actionneurs. Par exemple, le système cryogénique permettant de fournir aux aimants de l’hélium liquide régulé à 1.9K (-271.25°C) comporte plus de 40 000 entrées/sorties pour les automates chargés de son contrôle. Citons aussi le système chargé de réaliser le vide, permettant aux particules accélérées dans le LHC de circuler dans un univers « nettoyé » de toutes autres particules

considérées comme indésirables. Au niveau des expériences aussi, on retrouve outre les systèmes cryogéniques, d'autres systèmes tels que les systèmes gaz qui fournissent un mélange de gaz aux chambres composant les détecteurs.

1.1.1.3. Les opérateurs de contrôle

Le LHC fonctionne sans arrêt entre Février et Décembre de chaque année, car les opérations nécessaires à son démarrage ainsi qu'à son arrêt sont longues et complexes. Il est donc important de minimiser au maximum les arrêts techniques générés par d'éventuelles défaillances afin d'optimiser le temps de fonctionnement de l'accélérateur et augmenter ainsi le nombre de données récoltées.

Pour assurer le bon fonctionnement du LHC et de ses sous-systèmes, plusieurs équipes d'opérateurs se succèdent 24h/24, chacune ayant en charge la surveillance d'un processus particulier. Ils veillent à ce que les systèmes se trouvent bien dans la plage de fonctionnement pour lesquels ils ont été conçus et détectent tout comportement anormal. En pratique, leur rôle consiste à observer plusieurs variables physiques de façon régulière et à reconnaître qu'une des variables observées évolue de manière inhabituelle. Dans ce cas, ils effectuent le diagnostic du système, c'est à dire qu'ils analysent les informations disponibles afin de retrouver l'origine du défaut. Puis ils réalisent les actions correctives nécessaires.

Toutes ces opérations s'exécutent au travers d'une interface entre l'homme et la machine (IHM), qui présente à l'opérateur les différentes valeurs des variables à surveiller et lui permet d'agir directement sur les différents composants du système. Cette IHM est l'un des composants d'un système bien plus grand que l'on appelle le système de contrôle.

1.1.2. Le système de contrôle du LHC.

Au CERN, même si les systèmes sont interconnectés, chaque système de contrôle est indépendant des autres et agit de manière autonome, c'est-à-dire que chaque système possède sa propre architecture de contrôle qui ne dépend pas des autres systèmes. Il existe deux types de systèmes de contrôle au CERN : ceux dédiés à la production et à la conduite des faisceaux de particule basés sur des technologies propre à l'organisation. Et ceux qui contrôlent les procédés dit annexes (cryogénie, vides,...) ou les utilités (ventilation, refroidissement,...) qui utilisent des systèmes industriels, ce sont ces derniers qui nous intéressent ici.

1.1.2.1. Architecture de contrôle des procédés annexes.

Comme tout processus industriel, les procédés dit annexes du CERN utilisent des systèmes de contrôle continus qui permettent d'agir et surveiller les variables physiques du système. L'architecture des systèmes suit un schéma classique sur trois couches :

La couche terrain (Field layer) : Elle se trouve à proximité du processus, elle regroupe les bus de communication terrain qui permettent la communication avec les capteurs et les actionneurs (il s'agit d'un réseau robuste adapté au milieu industriel).

La couche contrôle (Control layer) : Elle est généralement composée de PC industriels ou d'Automates Programmables Industriels (API ou PLC en anglais). Elle contient les programmes permettant le pilotage du processus. C'est ici que les commandes à appliquer sur le système sont calculées, en fonction des informations mesurées par les capteurs. En général, c'est un algorithme informatique qui est exécuté de manière cyclique en temps réel. Cet algorithme lit les données du processus (entrées du système de commande), exécute le programme qui calcule les sorties du système de commande, puis transfère ces valeurs de sortie vers les actionneurs.

Couche de supervision (Supervision layer): elle permet de réaliser l'Interface Homme Machine. Au travers de cette interface les opérateurs surveillent les installations via des terminaux de supervision et peuvent agir sur la conduite du processus. Cette couche contient aussi une base de données qui archive les données des processus. Au CERN, le logiciel WinccOA© réalise l'interface Homme machine pour la plupart des systèmes annexes.

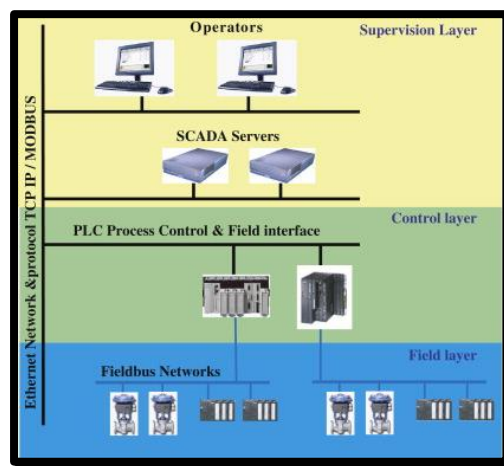


figure 1-2 Architecture de contrôle

1.1.2.2. Les données de système de supervision ou SCADA

C'est au niveau de la couche de supervision que les opérateurs interagissent avec le processus. Pour ce faire, les valeurs mesurées par les capteurs sont remontées à l'opérateur et affichées sur l'écran de l'Interface Homme Machine. Les ordres de l'opérateur sont transmis aux automates en suivant le chemin inverse.

Ces informations de natures différentes peuvent être classées en deux grandes catégories : les informations de nature continue dont les valeurs sont numériques et les informations de nature discrète dont les valeurs sont booléennes.

Les informations continues représentent les valeurs des variables physiques mesurées à l'aide d'un capteur approprié (températures, pressions débits ...). Ces valeurs sont affichées soit sous forme de valeur numérique instantanée soit sous forme de courbe de tendance

représentant leur évolution en fonction du temps. L'opérateur les inspecte régulièrement afin de s'assurer du bon fonctionnement du système.

Les informations discrètes disponibles informent les opérateurs de tout évènement ayant lieu au niveau du procédé, qu'il s'agisse d'une action demandée par l'automate ou par un opérateur, ou du changement d'état d'un module de commande (mise à 1 ou à 0 d'une sortie pour provoquer par exemple l'ouverture ou la fermeture d'une vanne, déclenchement d'une alarme,...).

Le système de contrôle déployé au CERN est composé d'un ensemble de sous-éléments (ou objets), organisés en hiérarchie. Chaque objet ne dépend que d'un supérieur hiérarchique (parent) mais peut agir sur plusieurs objets (enfants). Un objet peut être un objet de mesure (associé à un capteur), un objet de terrain associé à un actionneur, ou un objet de contrôle représentant un processus contrôlant plusieurs objets de terrain (PCO) ou des régulateurs.

Chaque objet reçoit :

- des informations venant du procédé (Valeurs analogique ou binaires des capteurs et états de ses objets enfants)
- des ordres de l'opérateur
- des ordres logiques de la part de son parent

L'objet de contrôle (PCO) exécute ses instructions en fonction des entrées reçues et modifie ses sorties (ordres donnés aux actionneurs ou aux PCO enfants).

Chaque objet informe la supervision ainsi que ses objets parents quand son état est modifié, et notamment s'il est en défaut. Il existe aussi des objets alarmes qui sont spécifiquement conçus pour déclencher certaines alarmes.

Parmi les évènements disponibles au niveau de la supervision, les alarmes sont les éléments de supervision associés aux situations anormales. Elles permettent de prévenir l'opérateur d'un état anormal. On peut dégager quatre grandes familles d'alarmes dans le système de contrôle du CERN :

- Alarmes associées à une variable physique :

Ces alarmes sont déclenchées lorsqu'une des valeurs physique du système se trouve en dehors des plages de fonctionnements pour lesquelles le système a été conçu. Elles sont donc toujours associées à des valeurs analogiques mesurées ou calculées. Ce sont des alarmes qui sont déclenchées lorsque la valeur dépasse un certain seuil évalué comme limite. Au CERN, il existe 4 seuils pouvant déclencher une alarme en fonction de la valeur de la variable. Le niveau « Bas » (« Low ») est déclenché lorsque la valeur de la variable descend sous un premier seuil fixée et le niveau « Très Bas » (« Low Low ») lorsqu'un second seuil plus bas est franchi. De façon similaire, les alarmes déclenchées peuvent correspondre à un niveau

« Haut » (« Haut ») et « Très Haut » (« High High »). La Figure 1-3 ci-dessous montre l'exemple d'une variable franchissant successivement les 4 valeurs seuils.

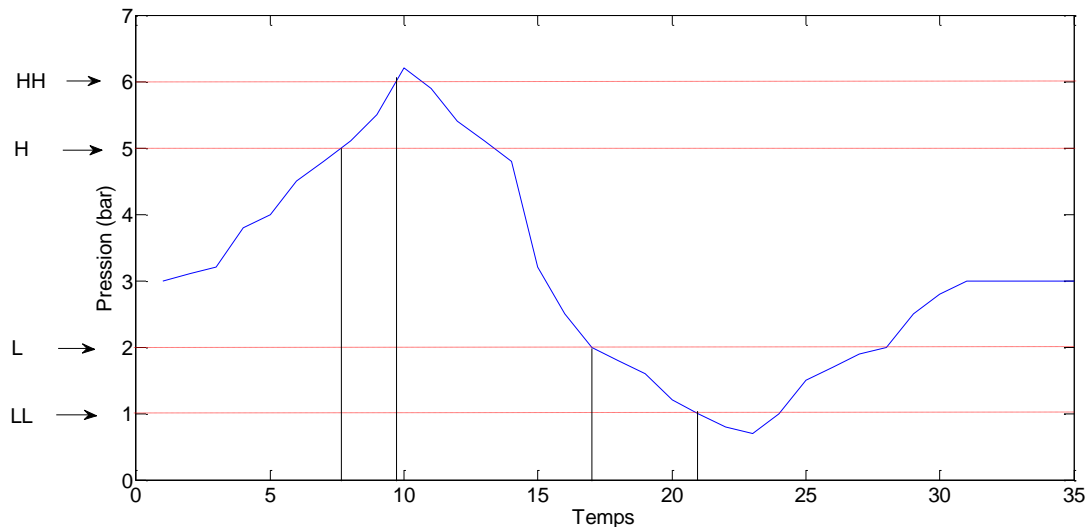


Figure 1-3 Exemple de déclenchement d'alarme sur franchissement d'un seuil

- Alarmes associées à une discordance :

Elles sont associées à des variables de type continu ou booléen, elles se déclenchent lorsque la valeur de la variable mesurée n'est pas celle qui est attendue. Dans le cas de valeurs continues, elles sont détectées lorsque l'écart entre la mesure et la valeur attendue est au-dessus d'un seuil limite. Pour des capteurs de type booléen, elles sont déclenchées lorsque la valeur du capteur est inverse de celle attendue (lorsque l'on donne l'ordre à une vanne de s'ouvrir et que le capteur de fin de course indique de la vanne n'est pas ouverte par exemple)

- Alarmes associées à l'état d'un actionneur :

Ces alarmes décrivent un défaut lié à un actionneur qui possède des éléments de mesure de son état de défaut. Comme, par exemple une pompe capable de signaler que sa température interne est trop haute.

- Alarmes liées à la structure du programme de commande :

Ce sont des alarmes propres aux systèmes de contrôle, elles fournissent des informations liées à l'état des différents objets du système de contrôle, comme les PCO. Par exemple, une alarme peut informer l'opérateur de l'arrêt d'un module.

L'ensemble des alarmes déclenchées sont transmises à la supervision à la fin du cycle automate, après exécution du code automate. Les alarmes arrivent donc par paquets à la supervision et sont horodatées par l'automate au même instant car elles sont déclenchées lors

du même cycle. On notera que le temps minimal entre deux paquets d'alarmes correspond au temps de cycle de l'automate.

Les alarmes sont retournées à la supervision sous forme de listes, avec certaines informations liées à l'alarme. On retrouve notamment :

- Son nom
- Une description de l'alarme
- L'heure d'occurrence de l'alarme
- Le composant ou le module concerné par l'alarme
- Une information sur l'état de l'alarme (active, inactive, acquittée, non acquittée)

Une alarme est activée quand elle passe de la valeur 0 à 1 i.e. elle se déclenche. Elle est désactivée quand elle passe de l'état 1 à 0 i.e. elle s'arrête. Elle est acquittée si l'opérateur à notifier sa prise en compte, non acquittée sinon.

1.2. Objectifs de la thèse

1.2.1. Difficultés de diagnostic des systèmes complexes

La complexité des systèmes du CERN ainsi que leur taille rendent la tâche de diagnostic difficile pour l'opérateur. Les alarmes générées par le système de contrôle alerte l'opérateur sur un dysfonctionnement probable du système mais ne lui fournissent pas l'origine du défaut, qui doit être identifié à partir des informations fournies par le système de supervision, et notamment les alarmes.

Lorsqu'un problème survient au niveau d'un des composants, l'origine du problème n'est pas facilement identifiable car le nombre d'alarmes générées est important. Cette multiplicité des alarmes est due à la forte corrélation qui existe entre toutes les variables physiques du système (pression, température, débit...). Lorsqu'une variable sort de ses limites de fonctionnement, elle entraîne avec elle le dépassement de seuil de toutes les variables avec lesquelles elle est corrélée, qui à leur tour entraîneront le dépassement de seuil des variables qui leur sont corrélées et ainsi de suite jusqu'à la propagation à tout le système, créant ainsi plusieurs centaines d'alarmes.

La corrélation physique n'est pas la seule responsable de cette avalanche d'alarmes, la logique de contrôle du système peut également contribuer à cette propagation. Par mesure de sécurité, le système de contrôle peut décider l'arrêt de certaines parties du système (un PCO) entraînant ainsi l'arrêt des sous-systèmes dont le fonctionnement dépend du premier sous système, qui déclenchent à leur tour une nouvelle série d'avalanche d'alarmes. Ces deux phénomènes peuvent s'enchaîner l'un après l'autre, déclenchant ainsi des centaines d'alarmes et pouvant arrêter le système dans sa totalité.

De plus, un même défaut ne déclenchera pas exactement la même avalanche d'alarmes. En effet, les différents états de fonctionnements dans lesquels se trouve le système sur l'occurrence du défaut, l'environnement extérieur qui varie, le bruit s'ajoutant sur les mesures font des systèmes du CERN des systèmes non déterministes, pouvant donc déclencher des alarmes différentes pour une même origine. Il est aussi important de noter que la première alarme ne donne pas nécessairement l'origine du problème.

La taille des systèmes, les fortes corrélations entre variables physique et logiques, ainsi que la variabilité dans les alarmes déclenchées rendent la tâche des opérateurs complexes. De plus, les informations disponibles (valeurs de variables et événements) ne fournissent pas de diagnostic à l'opérateur mais juste une information sur l'état d'un composant ou d'une variable. Le but de ce travail de recherche est d'aider l'opérateur, sur l'occurrence d'une avalanche d'alarmes, à retrouver le défaut premier en lui fournissant une analyse automatique et une interprétation des informations fournies par le système de supervision, dans le but de faciliter le diagnostic.

1.3. Formalisation du problème

1.3.1. Cahier des charges

Afin de pouvoir être applicable à l'ensemble des systèmes du CERN, la méthodologie développée doit satisfaire un certain nombre de contraintes :

Nous nous restreignons à l'utilisation des données binaires (alarmes) fournies par le système de supervision. L'ensemble des informations, continues et discrètes, étant trop important, nous souhaitons dans un premier temps évaluer les performances de méthodes à partir de données discrètes.

La méthodologie développée se doit d'être générique afin d'être applicable et adaptable à la plupart des systèmes du CERN. Or, ceux-ci sont nombreux et de natures différentes. Cela exclut donc de fait toutes les méthodes à base de connaissance (modèles, bases de règles), trop dépendantes du système. Nous opterons donc pour une méthode guidée par les données.

Nous considérerons qu'un seul défaut est à l'origine de l'avalanche d'alarmes et nous excluons le cas de défauts multiples. En effet, l'expérience des opérateurs montre que la probabilité d'occurrence de deux fautes indépendantes au même moment est relativement faible sur les systèmes du CERN.

Nous considérons que l'instant où l'analyse doit se déclencher est connu et correspond à l'apparition de la première alarme. En effet, le système de supervision est relativement stable. En fonctionnement normal, peu d'alarmes sont déclenchées et elles sont en général isolées. L'apparition d'une alarme rapidement suivie par plusieurs autres nous informe sur l'occurrence d'un défaut dont il faut rechercher l'origine. L'analyse n'est donc pas une

analyse au fil de l'eau mais est initiée par une alarme connue. Le diagnostic est fourni au bout d'un temps T à définir mais qui correspondra au plus tard au moment où le flot d'alarmes se tarit. (aucune alarme ne se déclenche plus après un certain délai d'attente).

1.3.2. *Approche adoptée :*

Afin de satisfaire ce cahier des charges, une approche guidée par les données a été choisie. Elle repose sur l'utilisation d'historique de données. Nous supposons que certains défauts ont déjà été observés plusieurs fois sur le système et que ces défauts ont été diagnostiqués par un expert. On dispose donc d'une base de défauts, chaque défaut étant représenté par la liste d'alarmes générée par le système de contrôle depuis le moment de l'apparition de la première alarme jusqu'à la stabilisation du flot d'alarme. Chaque alarme de cette liste sera considérée comme un symptôme du défaut.

Dans notre approche, nous supposons que l'ensemble des alarmes fournies par le système de contrôle est fini et connu. Chaque alarme est distincte des autres et prend une valeur binaire.

L'objectif du travail est de développer des méthodes capable de décider, sur l'occurrence d'un défaut à diagnostiquer, si le défaut fait partie des défauts listés dans la base ou non et, si c'est le cas, de reconnaître le défaut en particulier, où de proposer un ensemble de défauts probables. Nous nous attacherons à proposer des méthodes capables d'extraire de la connaissance des historiques de données disponibles.

2. État de l'art

Le diagnostic des systèmes complexes est un enjeu majeur dans le monde industriel de par les gains importants de temps et de productivité qu'il est possible de réaliser si le diagnostic est fait dans un délai court. La problématique a donc suscité de nombreux intérêts avec l'évolution des systèmes industriels. Mais, avant de nous intéresser aux différentes techniques de diagnostic développées dans la littérature, définissons ce qu'est un diagnostic. Le petit Robert le définit comme suit :

« Identification de la nature d'une situation, d'un mal, d'une difficulté, etc., par l'interprétation de signes extérieurs »

La définition met en lumière deux aspects du diagnostic. Le premier l' « *Identification de la nature d'une situation, d'un mal, d'une difficulté* » décrit en fait le but même du diagnostic, mais avec un champ d'application très large, les sujets que l'on cherche à identifier sont dans cette définition vaste : « *nature d'une situation, d'un mal, d'une difficulté* ». Cette identification s'appuie sur un processus d'interprétation, qui définit la méthodologie pour atteindre le but décrit auparavant. Mais ici le support de cette interprétation que sont les signes extérieurs reste assez vague. On va préciser ces termes dans le cas du diagnostic des systèmes complexes en s'appuyant sur les différents travaux de diagnostic de la littérature.

2.1. Définition du vocabulaire propre au diagnostic.

Dans le contexte du diagnostic de systèmes industriels, les termes liés à « *la nature d'une situation, d'un mal, d'une difficulté* » sont nombreux et leur signification varie en fonction de l'auteur et de la traduction faite du terme. On retrouve le plus souvent les termes de défaut, défaillance et panne. On propose de définir ces termes pour la suite du manuscrit de la manière qui suit :

- **Défaut/Faute** : Déviation non permise d'au moins une propriété caractéristique du système de ses conditions acceptables (Iserman & Balle, 1997)
- **Défaillance** : Une défaillance est l'altération ou la cessation de l'aptitude d'un ensemble à accomplir sa ou ses fonctions requise(s) avec les performances définies dans les spécifications techniques. L'ensemble est indisponible suite à la défaillance. Un ensemble est défaillant si ses capacités fonctionnelles sont interrompues. (Zwingelstein, 1995)

D'après les deux définitions, une défaillance est la résultante d'un défaut déclenché au niveau du système. L'inverse n'est cependant pas vrai, une défaillance n'impliquera pas forcément un défaut.

- **Panne** : La panne est l'inaptitude d'un dispositif à accomplir sa fonction vitale. Il est clair que, dès l'apparition d'une défaillance, caractérisée par la cessation du dispositif à accomplir sa fonction, on déclarera le dispositif en panne. Par conséquent, une panne résulte toujours d'une défaillance. (Zwingelstein, 1995)

On a donc une relation directe entre les trois termes avec différent niveau de criticité, un défaut est donc le premier signe d'un système dans un état anormal. La défaillance fait suite à un défaut, elle représente donc l'altération ou la cessation d'un ensemble à accomplir sa fonction, il sera donc considéré en état de panne. Dans la suite de ce document, on cherchera à identifier le défaut à l'aide des techniques de diagnostic développées dans le cadre de cette thèse.

Le diagnostic se base sur l'interprétation de signes extérieurs. Dans le domaine du diagnostic industriel comme dans le monde médical, ces signes sont appelées symptômes, on se référera à la définition faite par le dictionnaire « Reliability, availability, and maintainability » qui propose une définition du terme.

- **Symptôme** : Un changement d'une quantité observable de son état normal (RAM, 1988)

Dans notre cas, nous considérerons les alarmes comme les symptômes de notre processus de diagnostic. En effet, ils sont dans le cas des systèmes automatisés, l'expression d'un changement d'une quantité observable, comme le franchissement d'un seuil par exemple.

2.2. État de l'art des méthodes de diagnostic à l'aide d'informations binaires

Dans le domaine du diagnostic de systèmes complexes, on peut dégager deux grandes approches. La première est l'approche à base de connaissance, où le comportement du système est connu et est comparé à celui du système (de Kleer, 1987) (Reiter, 1987). La connaissance peut être représentée à partir de modèles ou de règles. Il peut s'agir de la connaissance du fonctionnement normal du système mais il peut également s'agir de la connaissance du fonctionnement d'un système en défaut. La seconde, celle qui nous intéresse dans ce travail de thèse, est l'approche utilisant un historique de données. On suppose que l'on dispose d'une base d'apprentissage à partir de laquelle il sera possible d'inférer des méthodes de diagnostic. Dans cette partie, après un bref tour d'horizon des techniques de diagnostic à base de modèle utilisant les données binaires, on se concentrera sur les méthodes de diagnostics à partir de données, en se focalisant plus particulièrement sur les alarmes comme source principale d'information.

2.3. Méthodes de diagnostic à base de connaissances

2.3.1. Les systèmes experts

Dans le domaine du diagnostic à l'aide de connaissances, les systèmes experts appartiennent au monde de l'intelligence artificielle (Liao, 2005). La connaissance est

retranscrite sous forme de règles simples et où le diagnostic est réalisé à l'aide d'un moteur d'inférence qui, à partir des règles, réalise le diagnostic à l'aide d'un raisonnement abductif. Ces règles peuvent être extraites à partir de la connaissance d'un expert du système. Des exemples d'applications de ces systèmes experts peuvent être vus dans (Al-Taani, 2007), (Qian, et al., 2003) et (Bernard & Duroche, 1993). Elles peuvent aussi être extraites de différentes sources comme les manuels de spécifications ou même de données provenant de la machine (Tan, et al., 2000). La limite de ces méthodologies est l'accès aux connaissances et aux règles. En effet, elles nécessitent une grosse implication de l'expert du système qui, la plupart du temps, n'est pas disponible. De plus, il est difficile d'exprimer les règles de façon exhaustive et le raisonnement peut parfois être opaque.

2.3.2. Diagnostic à base de modèles

2.3.2.1. Approche bon fonctionnement

Un modèle de bon fonctionnement représente le comportement du système sans défaut. Les entrées/sorties mesurées du système sont utilisées afin de les comparer à celles générées par le modèle. Tout écart entre les données et le modèle constitue un symptôme. Les méthodes avec un modèle de bon fonctionnement sont principalement utilisées dans le but de détecter l'apparition d'un défaut (Roth, et al., 2009). Les modèles utilisés peuvent être de différents types : les automates à états finis, par exemple, sont utilisés dans (Pandalai & Holloway, 2000) pour modéliser un système de convoyeur, les séquences d'événements doivent dans ce cas être respectées (avec des contraintes temporelles) pour considérer le système en bon état de fonctionnement. Sur le même principe, il existe de nombreux autres types de modèles dans la littérature parmi lesquels les plus utilisés sont les réseaux de pétri qui présentent l'avantage d'avoir une représentation graphique et mathématiques (Giua & Seatzu, 2001) (Hashtrudi Zad, et al., 1999) (Boufaied, 2003).

Bien que les modèles de bon fonctionnement soient majoritairement utilisés pour détecter un défaut, il existe dans la littérature des méthodologies qui proposent également de rechercher la source du défaut. (Reiter, 1987) propose une formalisation de la modélisation des systèmes et une modélisation à l'aide de relation logique du premier ordre. Le diagnostic est effectué en comparant le modèle avec les observations faites sur le système, l'identification du composant avec un fonctionnement anormal est déduite en recherchant quel composant a pu mener à cet état de défaut, en accusant tour à tour les composants et en comparant la sortie du modèle à l'état du système le diagnostic. Le diagnostic est réalisé de façon similaire lorsque des modèles MFM "Multilevel flow model" sont utilisés. Les MFM sont des modèles principalement utilisés dans le but du diagnostic introduit par (Lind, 1994). Le système est modélisé sous forme de « buts » et de « fonctions » du système, ces fonctions sont réalisées par des composants et elles sont reliées par des flots de matière, énergie, fluide.

2.3.2.2. Approche de mauvais fonctionnement

Le diagnostic peut également être réalisé à l'aide de modèles incluant les défauts. Dans ce cas, les séquences d'événements du système réel sont directement comparées aux modèles contenant les défauts. Comme avec les méthodes sans défaut, ces modèles peuvent prendre de nombreuses formes qui trouvent d'ailleurs leur équivalence dans ceux des modèles sans défaut. On propose ici une liste non exhaustive des différents types de modèles les plus communément utilisés dans la littérature :

- Le réseau de pétri est un outil de représentation efficace pour des systèmes à événements discret (Lefebvre & Delherm, 2007), (Ushio, et al., 1998) (Ramírez-Treviño, et al., 2007), (Cabasino, et al., 2007) utilise un modèle de réseaux de pétri incluant les défauts afin de développer un système de diagnostic. Le diagnostic est réalisé à partir des informations rapportées par le système SCADA sur les états des différents composants.
- (Sampath, et al., 1995) Propose une qui permet de définir à partir d'un modèle discret du système un modèle de faute appelé diagnostiqueur.
- L'arbre de défaillances donne une représentation du défaut à l'aide de combinaisons logiques d'événements. Le diagnostic est réalisé à partir des symptômes en parcourant l'arbre jusqu'au diagnostic final. Le lecteur peut se référer à (Hurdle, et al., 2005) , (Geymayr & Ebecken, 1995) pour des exemples de diagnostic à l'aide d'arbre de défaillances. Les arbres de défaillances présentent l'avantage d'être simple à lire et à comprendre. Cependant, il est difficile d'incorporer des contraintes d'ordre ou de temps dans le modèle d'arbre.
- Les chroniques sont une représentation du système à partir de séquences d'alarmes ou d'événements qui, à la différence des arbres de défaillance, peuvent prendre en compte des contraintes d'ordre et de temps entre les événements. Ici on modélise uniquement les défauts. Le diagnostic est réalisé en ligne : si une suite d'événements est reconnue comme une chronique de défaut alors le diagnostic du défaut est réalisé (Taisne, 2006). Leur principale utilisation se trouve dans les réseaux de télécommunications (Cordier & Dousson, 2000) ou au service web (Pencolé & Subias, 2009) . A noter qu'il existe des méthodes hybrides où les chroniques peuvent être extraites d'un modèle du système lorsqu'il est sous forme de réseau de pétri (Guerraz & Dousson, 2004) , ou la reconnaissance en ligne est faite à l'aide de réseau de Pétri dérivé du modèle de chronique (Pencolé & Subias, 2009).

2.3.2.3. Conclusion sur les approches à base de modèle

Nous avons vu que les modèles peuvent être de deux sortes, ils peuvent inclure ou non les défauts. Les modèles sans défaut sont majoritairement utilisés pour détecter un défaut

alors que les modèles avec défauts sont utilisées pour localiser la source du défaut. Le choix d'un type de modèle doit alors se faire en fonction de l'utilisation recherchée.

Bien que les techniques de diagnostic à base de connaissances trouvent leur application dans de nombreux domaines, l'un de leur principal inconvénient est justement l'acquisition de la connaissance. L'obtention d'un modèle du système est une tâche bien souvent difficile et longue à réaliser, voire impossible dans certain cas. De plus, ces méthodes sont très fortement liées au système et sont très sensibles à ses variations et ses évolutions. Le modèle doit être modifié à chaque changement du système. Il n'est pas aisé de développer des méthodologies génériques et chaque système nécessite son propre modèle. Ces méthodes ne peuvent donc pas s'appliquer à notre problématique du CERN, où les systèmes sont d'une grande complexité, variés et très nombreux.

2.4. Méthodes de diagnostic à base de données

2.4.1. Corrélation d'alarmes

Avant de nous intéresser au diagnostic à proprement parler, il est intéressant de regarder du côté des méthodes d'analyse d'alarmes qui sont une première étape avant le diagnostic et qui sont un moyen intéressant d'aider les opérateurs. Ici on cherche à travailler sur l'interface homme-machine et la présentation de l'information à l'opérateur se focalisent notamment sur la réduction du nombre d'alarmes présentées. Le problème de diagnostic peut être traité dès la phase de conception du système de supervision (Lambert, et al., 1999), (Hollifield, et al., 2008), ici il convient de bien choisir les alarmes qui seront présentes dans le système de supervision, et de travailler sur les méthodes de représentation de ces alarmes. On peut aussi étudier à posteriori les alarmes lorsqu'on possède un retour de connaissances. On regroupe sous le terme de corrélation d'alarmes les méthodes permettant une meilleure analyse et interprétation du flot d'alarmes (Gardner & Harle, 1996) (Jakobson & Weissman, 1993). L'objectif est de réduire la quantité d'informations proposée à l'opérateur en regroupant certaines alarmes (Julisch, 2001) ou en éliminant celles qui sont inutiles. Un travail de représentation peut aussi être associé au travail de suppression (Liu, et al., 2003) (S. Salah, 2013). La réduction du nombre d'alarmes peut être soit réalisée à l'aide d'un modèle du système, comme les modèle « MFM » (Larsson, 1994) (Tuszynski, et al., 2002). Ici, le modèle permet de déterminer les relations entre les alarmes à l'aide du modèle de flot et ainsi de hiérarchiser les alarmes en déterminant les alarmes qui sont les conséquences d'autres alarmes (elles ne seront alors pas présentées à l'opérateur). Des méthodes statistiques peuvent également être utilisées pour regrouper les alarmes (YAng, et al., 2012). Ces derniers proposent une représentation sous forme matricielle de la valeur des similarités des alarmes afin d'aider à leur regroupement. Enfin les méthodes de fouille de données permettent d'extraire de l'information des alarmes afin de déterminer les séquences fréquentes et ainsi réduire le nombre d'alarmes ou de donner des priorités aux alarmes (Hatonen, et al., 1996) (Kondo, et al., 2012) (Dousson & Vu Duon, 1999).

2.4.2. Diagnostic de défauts

On s'intéresse maintenant au diagnostic à proprement parler. On se focalisera dans ce paragraphe sur les méthodes à base de données binaires. Contrairement aux méthodes vues au paragraphe 2.3, les méthodes à base de données ne nécessitent pas l'élaboration de modèles. Elles reposent sur l'expérience acquise sur le système via des historiques de données. Les données peuvent être traitées de deux manières, en prenant en compte ou non l'ordre d'apparition des événements. On présentera, dans un premier temps, les méthodes n'utilisant pas les relations d'ordre dans les données avant de s'intéresser aux méthodes avec relations d'ordre.

2.4.2.1. Méthodes n'utilisant pas les relations d'ordre entre événements

L'ordre d'apparition des événements n'est pas considéré comme important pour ces méthodes. Les listes d'événements sont alors représentées sous forme de vecteurs binaires. Les méthodes les plus utilisées s'appuient sur des méthodes de reconnaissance de formes. La reconnaissance de forme consiste à catégoriser un élément inconnu appelé forme. Un vecteur forme est la description d'un individu à l'aide de caractéristiques le décrivant au mieux. La reconnaissance de formes consiste à attribuer une classe à la forme inconnue. Dans le cas du diagnostic, les classes représentent généralement un défaut à reconnaître, et le vecteur forme est l'ensemble des alarmes. Cette méthodologie présuppose que l'on dispose d'une base de données (appelée base d'apprentissage) contenant des observations passées faites sur le système. A partir de ces données récoltées, on apprend automatiquement la fonction de décision qui associe une classe à un vecteur forme, l'algorithme réalisant la classification est appelée classifieur. On peut scinder l'apprentissage en deux grands groupes. Dans le premier appelé apprentissage non supervisée, la base d'apprentissage est composée d'exemples dont la classe est inconnue. Dans ce type d'approche, il s'agit de déterminer les classes en regroupant les exemples d'une même population. Le second correspond à l'apprentissage supervisé où les exemples présents dans la base d'apprentissage sont correctement classés par un expert. Dans ce cas, le problème consiste à apprendre une fonction de décision capable de prendre la même décision que l'expert. On présente dans ce qui suit des exemples de diagnostic à partir de méthodes de reconnaissance de formes utilisant la classification supervisée. Les alarmes produites sur l'occurrence d'un défaut sont stockées dans un vecteur forme.

Réseau de neurones : L'une des approches populaires est le réseau de neurones artificiels et notamment le perceptron multi-couches (PMC). Un PMC est composé de plusieurs couches de neurones interconnectés. Les neurones de la première couche sont activés par les caractéristiques de l'observation, cette information est alors passée, après pondération et transformation, aux neurones de la deuxième couche, ainsi de suite jusqu'à ce que les neurones de sortie qui déterminent la classe soient activés. La structure du PMC ainsi que le nombre de couches sont à déterminer, les fonctions de pondérations sont-quant à elles apprises à l'aide des données lors de la phase d'apprentissage. (Al-Jumah & Arslan, 1998) présente une méthodologie de diagnostic basée sur l'utilisation d'un PMC à 3 couches

appliquée à du diagnostic de circuits électriques (Chen, et al., 2011) utilise un PMC pour diagnostiquer des défauts sur des éoliennes à partir des alarmes SCADA et (Negnevitsky & Pavlovsky, 2005) pour diagnostiquer des relais de protection dans des circuits électriques. Bien que les réseaux de neurones constituent des outils puissants de classification, ils présentent quelques inconvénients. Leurs performances dépendent de la structure utilisée, et le résultat est difficilement interprétable car la logique de raisonnement menant au résultat n'est pas explicitée.

Induction d'arbres de décision : Les arbres de décision sont une représentation graphique d'un ensemble de règles de type si-alors menant à une décision. Les feuilles de l'arbre représentent une décision (un défaut, par exemple), les nœuds de l'arbre représentent un test sur une caractéristique et les branches le résultat du test. L'induction d'arbres de décisions construit de façon automatique un arbre de décision à partir d'un ensemble d'apprentissage. Dans (Charkaoui, 2005), des arbres de décision sont construits afin de réaliser la détection et la localisation de défaillances multiples à partir des logs d'alarmes stockées dans l'ordinateur de bord de véhicules. L'apprentissage automatique de fonctions de décisions présente l'avantage de ne nécessiter aucun modèle et de reposer uniquement sur des historiques de données disponibles sur le système. Cependant, la phase d'apprentissage est très gourmande en données.

Une alternative à ces méthodes repose sur des méthodes de classification à partir de mesures de distances ou de similarités (pattern matching) où un classifieur de type « k plus proches voisins » est utilisé. Dans (Yemini, et al., 1996), les défauts dans des réseaux de télécommunications sont décrits sous forme de vecteur binaires au chaque élément du vecteur représente un symptôme du défaut et le diagnostic est réalisé par comparaison entre le défaut à diagnostiquer et des précédentes observations à l'aide d'une distance de Hamming. D'autres distances plus complexes peuvent également être utilisées comme dans (Charkaoui, et al., 2005). L'intérêt de ces approches par distances ou mesures de similarité est qu'elles peuvent être mises en œuvre avec un nombre restreint d'exemples par classe.

2.4.2.2. Méthodes utilisant la relation d'ordre entre évènements

Les méthodes présentées au paragraphe précédent supposent que l'ordre d'apparition des alarmes n'a pas d'importance pour réaliser le diagnostic. Cependant, c'est une information supplémentaire qui peut être utile au diagnostic. Dans ce cas, les listes d'alarmes sont représentées par une séquence ie par un ensemble ordonné d'évènements ou d'alarmes où le temps séparant l'apparition de deux évènements peut aussi être pris en compte. Ces séquences peuvent correspondre à des séquences de bon fonctionnement (Budalakoti, et al., 2009) mais également à des séquences de défauts (Cheng, et al., 2013).

Une première approche consiste à construire un modèle du système de manière automatique à partir des données puis à appliquer les mêmes algorithmes de diagnostic que lorsque le modèle est construit par la connaissance. Les modèles appris sont similaires à ceux vu au paragraphe 2.3. Ce sont, par exemple, des chroniques représentant un défaut par une

séquence d'alarmes avec contraintes temporelles. Dans (Duong, 2001), (Dousson & Duong, 1999) la recherche de chroniques de défauts dans un réseau de télécommunication repose sur l'analyse de la fréquence d'apparition d'alarmes successives dans les différents logs d'alarmes. Un réseau de pétri peut également être appris d'un jeu de données d'alarmes (Lefebvre & Leclercq, 2011). D'autres modèles construits de manière empirique peuvent aussi être utilisés, comme les réseaux Bayésiens. Un réseau Bayésien représente de manière graphique les relations de cause à effet entre deux événements. Le modèle est constitué de nœuds et d'arcs (Charniak, 1991), un nœud représente une variable et les arcs les probabilités conditionnelles entre variables. Les probabilités associées à chaque arc sont apprises à partir des données. La décision est prise en calculant le défaut le plus probable à partir des données disponibles (Chen, et al., 2012), (Yamaguchi, et al., 2012). D'autres méthodes reposant sur des graphes probabilistes peuvent aussi être envisagées, comme les chaînes de Markov (K. Yamanishi, 2005), les modèles de Markov cachés (Ying, et al., 2000) ou les réseaux de raisonnement abductif (Sun, et al., 2012) (Huang, 2002). Ces méthodes, largement appliquées dans le domaine du diagnostic à partir de données binaires, présentent l'avantage de construire de façon automatique des modèles de défauts. Cependant, comme toutes méthodes d'apprentissage automatique, leur utilisation suppose qu'un grand nombre de données est disponible.

De manière similaire, des approches type fouille de données peuvent être envisagées (Mannila, et al., 1997). Ces approches utilisent des outils de recherche de motifs séquentiels fréquents introduits dans (Agrawal & Srikant, 1995). Les travaux utilisent une version adaptée de l'algorithme de recherche de motif fréquent (GSP) présenté dans (Srikant & Agrawal, 1996). Dans (Hostalkova & Stluka, 2010) et (Costa, et al., 2009), une variation de l'algorithme GSP est proposée dans le but de rechercher des motifs d'alarmes fréquentes dans un but de suppression d'alarmes ou de diagnostic. Bien que ces méthodes trouvent leurs applications avec succès dans (Folmer & Vogel-Heuser, 2012), elles présentent quelques limitations : elles ne sont pas robustes aux variations dans les séquences d'alarme. (Folmer & Vogel-Heuser, 2012) précise qu'une seule alarme qui diffère peut mettre en défaut le système. De plus, la recherche de motifs séquentiels reposant sur des propriétés statistiques, notamment des calculs de probabilités conditionnelles, elles nécessitent également une grande quantité d'information.

Une alternative aux méthodes d'apprentissage automatique est la comparaison de séquences d'événements qui utilisent des mesures de similarité entre deux séquences. Cela nécessite une mesure de similarité qui permet de chiffrer la ressemblance entre deux séquences. La similarité entre séquences est utilisée dans de nombreux domaines. Une application concerne la détection d'anomalies dans une séquence d'événements discrets (Chandola, et al., 2012). Dans ces situations, le jeu de données est constitué de séquences de bon fonctionnement et l'on cherche à déterminer si la séquence à analyser est conforme à ce bon fonctionnement. Les applications concernent la détection d'intrusions dans un réseau informatique (Lu, et al., 2005), (Hofmeyr, et al., 1998), mais on trouve également un exemple portant sur l'analyse du comportement de pilotes d'avions durant un atterrissage (Budalakoti,

et al., 2009). La mesure de similarité entre séquences concerne aussi la recherche de mots dans un texte. On peut faire de la recherche exacte c'est-à-dire que l'on recherche exactement la même séquence de lettre (Knuth, et al., 1987), (Karp & Rabin, 1987), ou l'on peut faire de la recherche approximative afin de déterminer par exemple des fautes de frappe. Un grand nombre d'algorithmes est disponible dans ce domaine (Navarro & Baeza-Yates, 1999) (Wu & Manber, 1992). Enfin, la Bio-informatique est un domaine où la comparaison de séquences est très utilisée pour comparer des séquences de gènes. Dans ce cas, on ne recherche pas des séquences exactes mais des séquences similaires. Dans ces procédures, les séquences sont dans un premier lieu alignées en ajoutant ce que l'on appelle des « gaps » dans l'une et l'autre séquence afin de maximiser les concordances entre les éléments de la séquence et enfin une mesure de similarité est effectuée (Smith & Waterman, 1981) (Needleman & Wunsch, 1970). (Cheng, et al., 2013) ont proposé d'utiliser l'algorithme de Smith et Waterman pour faire du clustering d'avalanches d'alarmes et déterminer si les clusters formés pouvaient être associés à des défauts particuliers.

2.5. Positionnement

Au regard du cahier des charges défini dans le chapitre 1, les approches à base de modèles ne font pas partie des possibilités qui s'offrent à nous. En effet, nous cherchons à développer une méthode de diagnostic qui puisse être générique et s'appliquer à tous les systèmes du CERN. De plus, développer un modèle pour des systèmes aussi complexes que ceux du CERN s'avère être une tâche longue, difficile voire impossible. Les exemples de défauts dont on dispose sont peu nombreux, ce qui exclut ainsi les méthodes d'apprentissage automatiques, que les données soient représentées sous forme de vecteurs ou de séquences, ces méthodes étant gourmandes en données. On se tournera donc vers des méthodes de pattern matching, où la liste d'alarmes à diagnostiquer est comparée via des mesures de distances ou de similarité à des exemples de défauts connus. On présentera deux algorithmes de diagnostic. L'un est basé sur une représentation vectorielle des listes d'alarmes et utilise une distance originale fondée sur un calcul de poids mettant en valeur les alarmes les plus pertinentes pour le diagnostic du défaut. Le second présente une approche où les listes d'alarmes sont représentées comme des séquences. Des outils dérivés de la bio-informatique sont utilisés à des fins de comparaison de séquences. Notons que (Cheng, et al., 2013) présentent des outils similaires à ceux proposés dans cette thèse. Ils utilisent un algorithme de bio-informatique pour regrouper des avalanches d'alarmes au sens d'une similarité mais ne proposent pas de méthode de diagnostic ni d'aide à l'opérateur. La publication de ces travaux a été effectuée au printemps 2013, alors que le travail de recherche de cette thèse était en phase de finalisation et que les choix méthodologiques avaient déjà été réalisés.

3. Simulateur des systèmes gaz

Dans ce chapitre, on présente le système choisi pour valider les méthodologies qui seront développées dans les prochains chapitres de ce document. Il a été décidé dès début du travail de thèse de développer un simulateur d'un système réel du CERN afin d'en extraire les données nécessaires aux validations des méthodes au lieu de reprendre les données d'un banc de test déjà existant, notre objectif étant de répondre à la problématique du CERN et de valider les méthodes en étant au plus proche des problèmes de diagnostic réel. De plus, il existe au CERN des outils puissants de simulation qui permettent le développement rapide d'un simulateur. Un autre avantage inhérent à l'utilisation d'un simulateur est la possibilité de remplir la base d'apprentissage et de disposer de données sans attendre que les défauts arrivent sur le système gaz réel.

3.1. Étude de cas : le système gaz

Les procédés sont multiples au CERN, la cryogénie, le système de mélange de gaz, les systèmes de ventilation etc.... Le choix s'est tourné vers les systèmes gaz car il s'agit des systèmes qui nécessitent le plus d'aide au diagnostic. Ils sont surveillés par des opérateurs non experts du système, qui sont plus des utilisateurs du système que de réels opérateurs.

3.1.1. Chambre Multifilaire Proportionnelle

Afin de comprendre à quoi sert un système gaz, intéressons-nous tout d'abord au principe de fonctionnement de la Chambre Proportionnelle Multifilaire « MPM » (Multi-wire proportional chamber). La MPM est un détecteur de particules inventé et développé par Georges Charpak au CERN en 1968 (Charpak, 1969) permettant un traitement informatique des données.

Une Chambre Multifilaire Proportionnelle est composée d'une enceinte étanche remplie d'un gaz (ou un mélange de gaz) noble. Des fils électriques parallèles sont tendus à l'intérieur de la chambre sur un même plan appelé anode. De part et d'autre de ce plan se trouve un plan chargé positivement appelé cathode. Lorsqu'une particule traverse la chambre, elle ionise le gaz en séparant les atomes en électrons chargés négativement et en cations chargés positivement. Les électrons sont alors attirés par l'anode et les cations par les cathodes, créant alors une impulsion électrique au niveau des cathodes qui permet de détecter le passage d'une particule et de le reconstruire à l'aide d'un traitement informatique. L'utilisation des chambres à fil a permis la multiplication par 1000 de la vitesse de traitement des données par rapport à la génération de détecteur précédente.

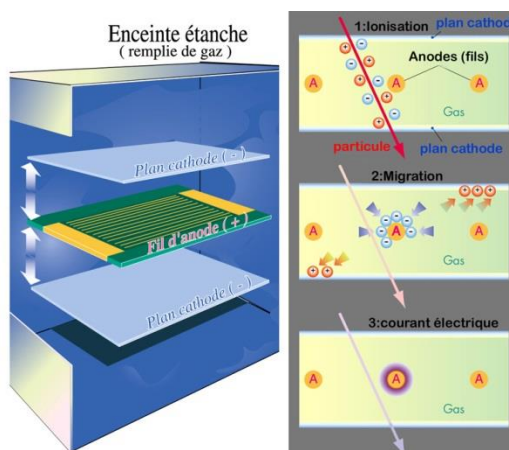


Figure 3-1 Représentation de la chambre multifilaire proportionnelle

3.1.2. Le « système gaz »

Le Système à Gaz ou « Gas system » en anglais a pour but de fournir un mélange de gaz à la Chambre Multifilaire Proportionnelle en respectant scrupuleusement les proportions des différents gaz, tout en s'assurant de la pureté de ce mélange. Son autre rôle consiste à réguler de façon précise la pression à l'intérieur d'une chambre de mesure. Pour ce faire, tous les systèmes gaz sont construits selon le schéma qui suit :

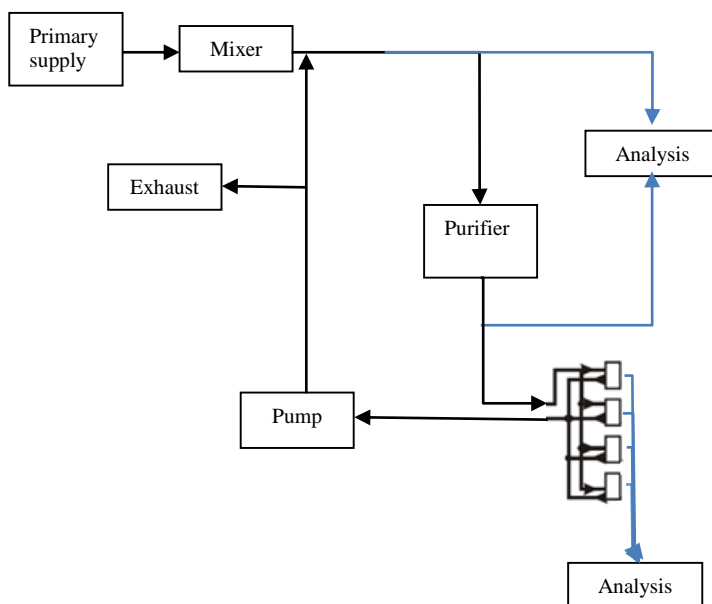


Figure 3-2 Vue d'ensemble d'un système gaz

Le système gaz est une boucle fermée où un mélange de gaz circule. Chaque système gaz se décompose en 7 différents modules :

- Module Mixeur (Mixer)

- Module Distribution
- Module Pompe (Pump)
- Module Purification (Purifier)
- Module Echappement (Exhaust)
- Module Analyse de gaz (Analysis)
- Module Récupération (Recovery)

3.1.2.1. Module Mixeur

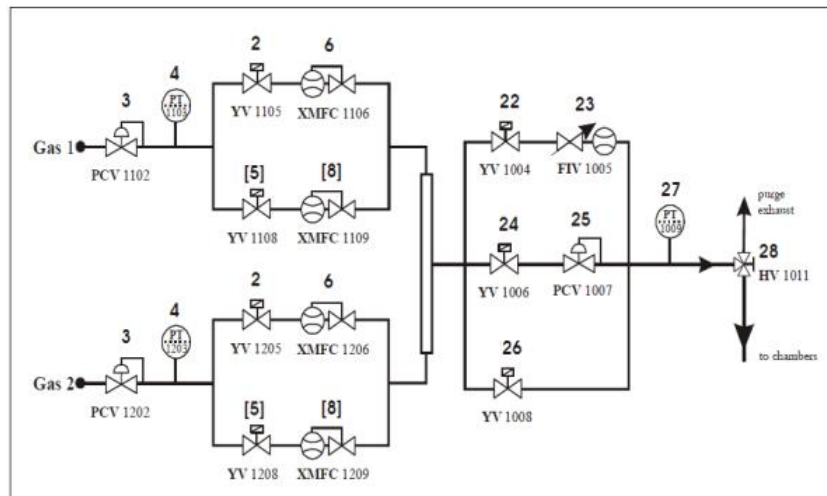


Figure 3-3 Représentation du Module Mixeur

Le module mixeur sert à mélanger les gaz purs afin de créer le mélange de gaz. Les systèmes gaz peuvent mélanger entre 2 (identique au schéma ci-dessus) et 3 gaz. Les gaz sont mélangés dans le volume se trouvant au centre de l'image, le mélange se fait de façon passive dans le mixer, sans pièce mobile à l'intérieur de celui-ci. Le ratio de gaz est assuré à l'aide d'un appareil appelé « Mass Flow Controller » « MFC », il s'agit d'une vanne disposant d'un contrôleur PID intégré. À la sortie du module, la pression est également régulée.

3.1.2.2. Module Distribution

Le module distribution sert à alimenter en gaz les différentes chambres qui composent le détecteur. À l'entrée du module distribution, la ligne principale se décompose en plusieurs racks qui chacun alimente une chambre multifilaire. En sortie des chambres, les flots de gaz se rejoignent à nouveau dans la boucle principale du système gaz pour suivre le chemin dans la boucle principale. Les chambres étant relativement fragiles, la pression à l'intérieur de celles-ci doit être très précisément régulée autour de la pression atmosphérique.

On observe un phénomène de diffusion à travers les parois de la chambre entre le mélange de gaz et l'air, rendant ainsi le gaz impur.

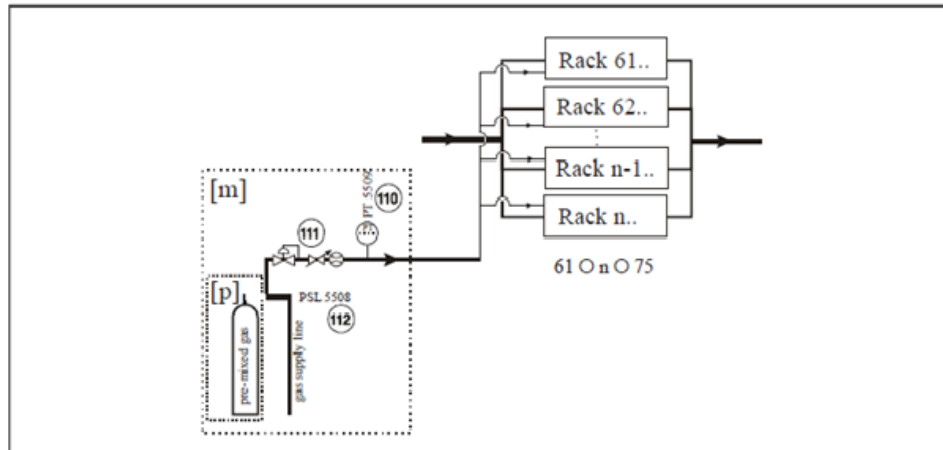


Figure 3-4 Schéma de racks

3.1.2.3. Module Pompe

Le module pompe est composé de 2 ou 3 pompes selon la configuration du système. Il permet de faire circuler les gaz dans la boucle. En fonctionnement normal, une seule pompe fonctionne. Un système de bypass avec une vanne régulée en pression permet de maintenir la pression en sortie du module pompe.

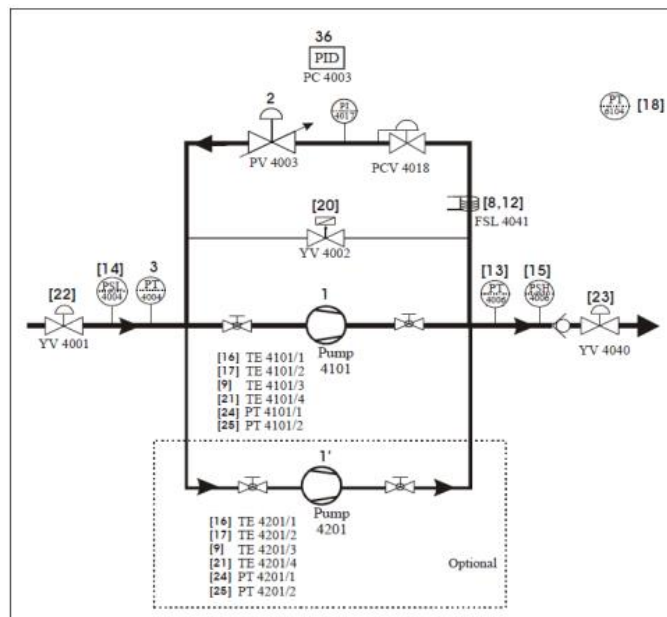


Figure 3-5 Schéma du module Pompe

3.1.2.4. Module Purification de gaz

Le module de purification a pour but de purifier le gaz qui circule dans la boucle. Des impuretés s'accumulent dans le gaz en circulation notamment au niveau de la chambre multifilaire proportionnelle dû à un effet de diffusion dans la chambre. La diffusion introduit

de l'air (principalement du CO_2 , N_2 et O_2) dans le mélange de gaz et de son côté le gaz diffuse au travers des parois de la chambre.

Le module de purification est composé de deux volumes remplis de charbon actif ayant pour but de filtrer les molécules de dioxygène et de dioxyde de carbone. Ces volumes sont utilisés en alternance car ils nécessitent un temps de régénération durant lequel ils sont inutilisables. Pendant la régénération d'un volume, le second filtre les impuretés du gaz. Le circuit du gaz est relativement simple : il arrive par le circuit principal, traverse le volume de purification et repart vers le circuit principal en sortie du module.

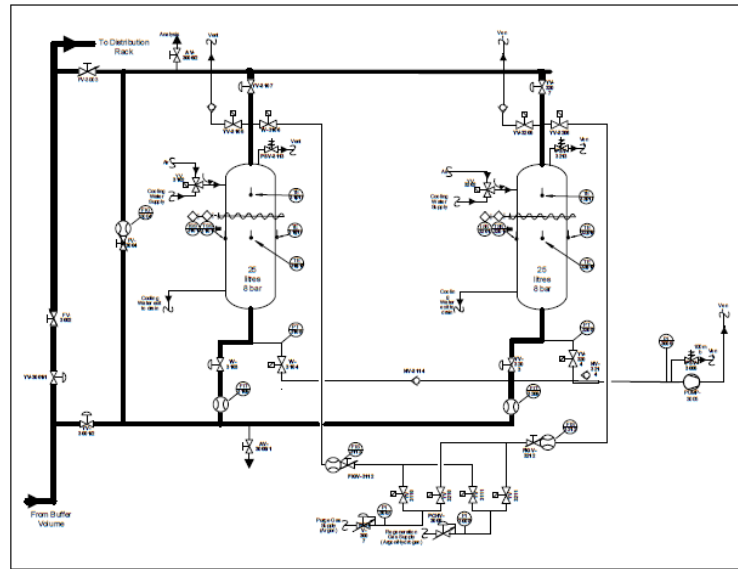


Figure 3-6 Schéma d'un module de purification

3.1.2.5. Module Analyse

Le module analyse mesure la qualité du mélange en plusieurs points du système gaz. Il extrait le gaz qui circule et l'analyse grâce à différents types de capteurs.

3.2. Modélisation du système

3.2.1. Ecosimpro

Ecosimpro est le logiciel utilisé au CERN pour développer le simulateur (Bradou, 2010). Il repose sur la modélisation de chaque composant du système. Les modèles sont des relations d'entrées- sorties pour les grandeurs physiques utiles dans le cadre du simulateur. Il possède son propre langage de programmation appelé « Ecosimpro langage ». Ce langage est particulièrement adapté à l'écriture d'équations mathématiques, notamment les équations différentielles. Il s'agit d'équation 0D ou 1D c'est-à-dire d'équation algébrique ou d'équations différentielles ne dépendant que d'une seule variable, le temps dans notre cas. On négligera tous les effets spatiaux.

La création du système complet se réalise grâce à la liaison des différents composants. Ecosimpro comporte un solveur d'équation, qui se charge de résoudre les problèmes de causalité du calcul ainsi que de résoudre le système d'équations. Tous ses composants sont décrits et regroupés dans une bibliothèque. On décrit, dans le paragraphe suivant, les modèles de chaque composant réalisés au cours de cette thèse pour modéliser le procédé, ainsi que les hypothèses faites lors de la modélisation. Etant donnée la méthode de résolution du logiciel, il est impossible de décrire l'ordre d'exécution des calculs, cependant les composants construits doivent suivre la procédure suivante.

Le composant est construit dans cet ordre :

- La définition des ports du composant : On définit dans un premier temps le nombre de ports et leur type.
- La définition des paramètres précise les différents paramètres caractéristiques du composant, fixé par l'utilisateur. Il s'agit de constantes propres au composant.
- La définition des variables précise les variables internes du composant.
- La phase d'initialisation du composant fixe les variables aux valeurs initiales définies au préalable.
- Calcul des propriétés physiques ; Les propriétés nécessaires à la définition du composant sont calculées tel que la densité, la température...
- Calcul des équations du composant : Les calculs sur les variables et les relations d'entrées-sorties sont effectués ici.

3.2.2. *Création d'une bibliothèque de composants du système Gaz*

La création du système complet se réalise grâce à la liaison des différents composants modélisant la physique de ceux-ci. Tous ces composants sont regroupés dans une librairie. Les modèles de chaque composant ainsi que les hypothèses faites lors de la modélisation se trouvent en annexe.

Un premier travail a consisté à vérifier, à l'aide de prélèvements des différents gaz, que l'hypothèse des gaz parfaits était vérifiée. (voir annexe).

Puis trois types de composants ont été modélisés :

- Les composants capacitifs : ils possèdent un volume interne, ils calculent une pression ainsi que les concentrations des différents composants du gaz en fonction des débits d'entrées et de sorties. Les volumes ou buffers, les tuyaux etc.... sont des exemples de composants capacitifs.
- Les composants résistifs : ils calculent un débit en fonction des pressions d'entrées sorties. Les vannes, les pompes sont des exemples de composant résistifs

- Les composants de contrôle. Il s'agit des composants associés au système de contrôle du gaz système. Les PID sont des exemples de composant de contrôle.

Afin d'avoir un système cohérent les nombres d'équations et d'inconnus doivent être strictement égaux. Chaque composant possède des conditions aux limites nécessaires au calcul de l'équation différentielle modélisant la relation entre l'entrée et la sortie du composant. Pour qu'il existe toujours une solution convergente à chaque pas de calculs il faut toujours alterner un composant capacitif avec un composant résistif

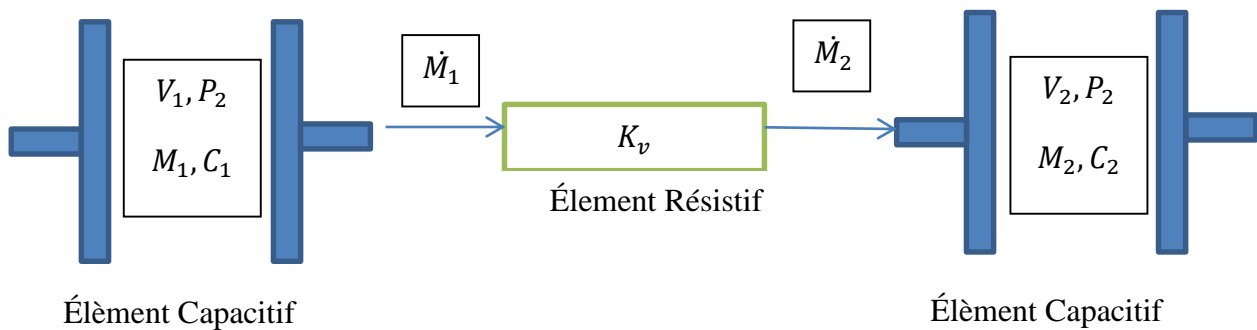


Figure 3-7 Représentation de l'alternance entre composants résistifs et composants capacitifs

A titre d'illustration, on détaillera la modélisation d'un composant capacitif appelé Buffer. Pour le détail de modélisation des autres composants, le lecteur pourra se référer à l'annexe. Pour information, un total de 9 composants principaux et une quarantaine de composants dérivés a été modélisé et forme la librairie du système gaz.

Les buffers sont l'un des principaux éléments capacitifs ils consistent en un gros volume permettant le stockage de gaz. On considère ici la modélisation d'un buffer de forme cylindrique. Il possède un port d'entrée type gaz et une sortie de type gaz. Il s'agit d'un composant passif sans aucun élément de contrôle. Le modèle repose principalement sur la loi des gaz parfait. Le volume du buffer est considéré comme constant (hypothèse isochore), et les échanges de température avec l'extérieur sont considérés comme nuls.

On définit dans un premier temps les variables utilisées pour modéliser le système :

m : Masse de gaz à l'intérieur du composant kg

$C[j]$: Concentration massique du gaz j

P : Pression en bar

MW : Masse molaire du gaz $kg \setminus mol$

V : Volume du composant en m^3

W : Débit massique en $\frac{kg}{s}$

Équation différentielle :

- Variation de masse en fonction des débits d'entrée et de sortie

$$\dot{m} = W_{in} - W_{out} \quad 3-1$$

- Les concentrations de chaque composant sont issues de l'équilibre des masses pour chacun d'eux :

$$\dot{C}[j] = W_{in} * \frac{C_{in}[j] - C[j]}{1 = \sum C[j]} \quad 3-2$$

- Pression du gaz :

Loi des gaz parfaits peut s'exprimer de la manière suivante :

$$P = m \cdot R \cdot \frac{T}{MW \cdot V} \quad 3-3$$

donc la variation de pression peut être définie par :

$$\dot{P} = \dot{m} \cdot R \cdot \frac{T}{MW \cdot V} \quad 3-4$$

3.2.3. Construction du modèle complet par liaison des composants

Afin de construire à l'identique les systèmes, on s'appuie sur le diagramme P&ID (Process and Instrumentation diagramme) du procédé, qui est le schéma descriptif du système. On réalise les liaisons entre chaque composant du système en entrant les paramètres adéquats, et en veillant toujours à respecter la règle d'alternance entre un composant capacitif et un composant résistif. Ainsi, il est aisé de construire chaque module en effectuant les différentes liaisons entre les composants, puis relier chaque module pour construire le modèle complet.

A titre d'exemple, la Figure 3-8 représente le diagramme P&ID du module purification.

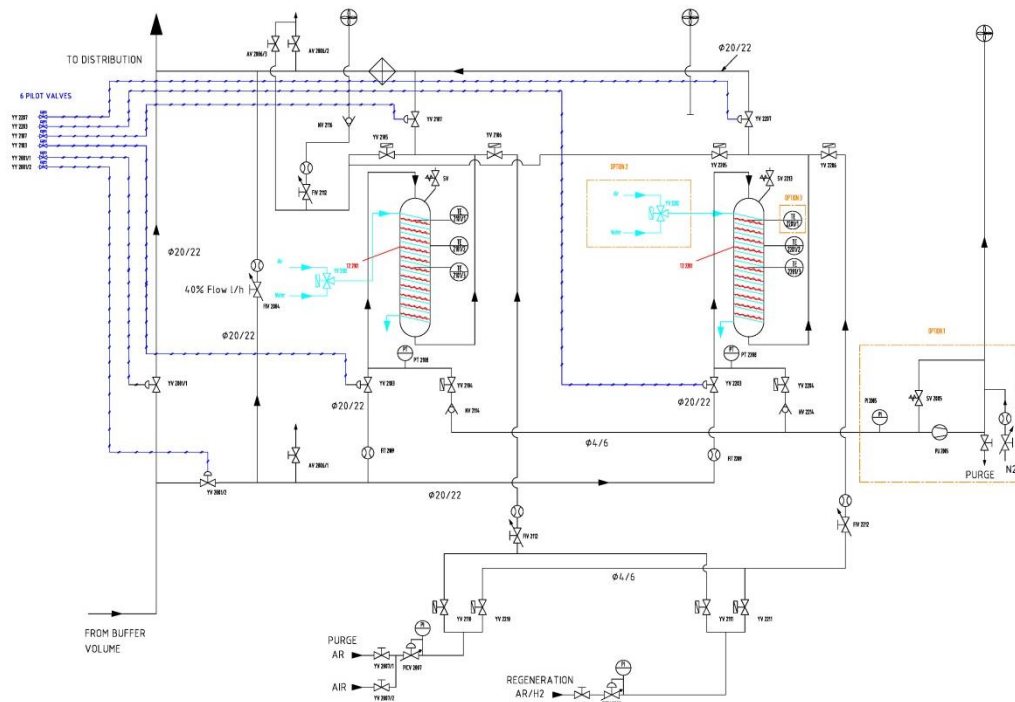


Figure 3-8 P&ID représentant le Module Purification

La version du module Purification dans Ecosimpro est la suivante :

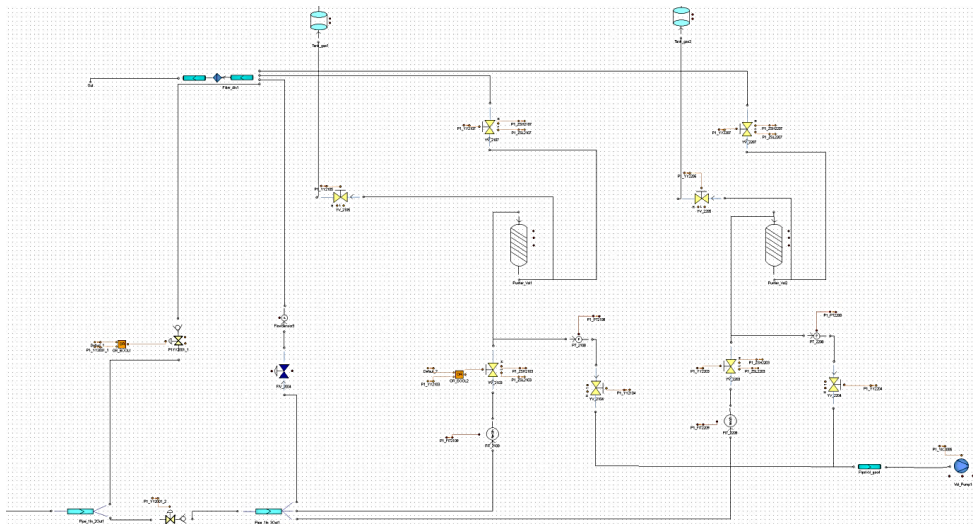


Figure 3-9 Modèle du module purification sous EcosimPro

3.2.4. Intégration au système de contrôle

Afin d'obtenir un simulateur complet, c'est-à-dire contenant toutes les différentes couches d'un procédé contrôlé classique, on intègre le modèle physique développé sous EcosimPro avec le système de contrôle. Il est ainsi possible de recréer l'environnement de contrôle du procédé. Le principe choisi est de garder l'ensemble des composants du système

contrôle et de remplacer le vrai système gaz par le modèle du procédé. La figure ci-dessous montre le schéma classique d'un système de contrôle connecté au procédé réel et son équivalent connecté au simulateur.

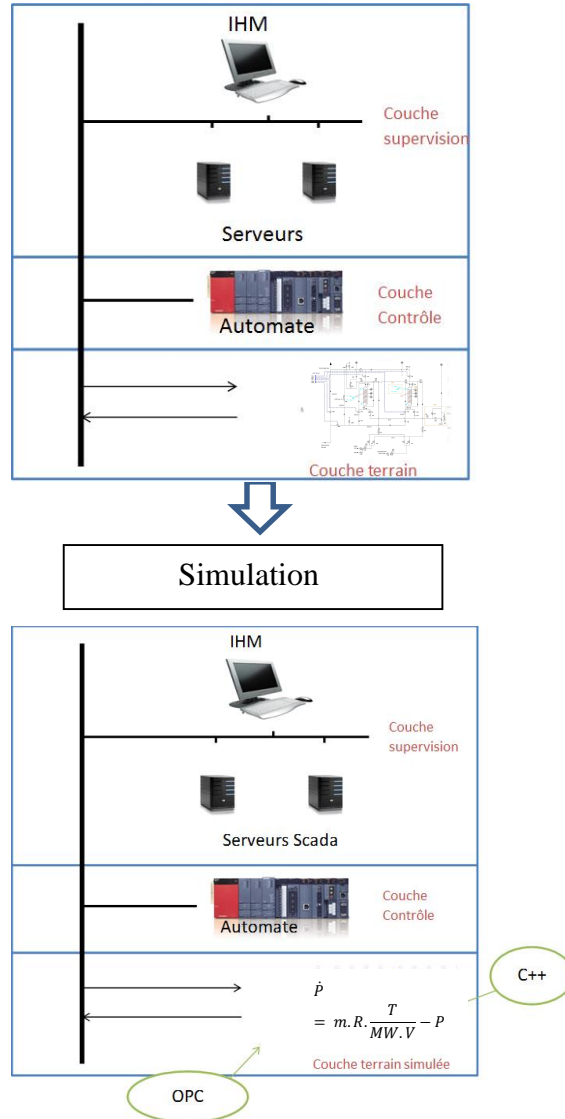


Figure 3-10 Intégration du modèle dans le système de contrôle

On peut voir que les couches contrôles (automates programmables) et supervision sont exactement les mêmes, que ce soit pour le système en production ou le simulateur. Le procédé réel a été remplacé par le modèle. Afin de permettre l'intégration de ce modèle dans l'architecture de contrôle, un code C++ est généré à partir du modèle développé sur EcosimPro. Les connexions physiques (câbles, bus de terrain) entre l'automate et les composants sont eux remplacés par un Server OPC permettant la communication entre l'automate et le modèle C++. Le code automate est le même qu'en production avec de légers changements pour réaffecter les adresses des entrées/sorties de l'automate, vers les adresses du code C++. Le système SCADA est lui inchangé. Concrètement, le simulateur est utilisé

comme le véritable système, à partir de la supervision : l'opérateur donne des ordres au système et l'automate transmet ses instructions au modèle C++, le modèle calcule la valeur de toutes les variables physiques et les transmet à l'automate à travers les différents capteurs simulés et les remonte à la supervision. Le temps de simulation varie en fonction de l'état du système simulé, lors de phase la transitoire (un démarrage par exemple) la vitesse de simulation est légèrement inférieure au temps réel et peut monter jusqu'à plus de 10 fois la vitesse réelle lors de la phase de stabilité (mais peut être ralentie si besoin). Ceci permet une utilisation proche du système réel du simulateur.

3.2.5. *Simulation des défauts et extraction des listes d'alarmes*

Les défauts sont directement intégrés dans le modèle, on peut facilement les déclencher à l'aide d'une variable automate booléenne. Le défaut est alors transparent pour le système de contrôle. Par exemple, dans le modèle d'un buffer une sortie de gaz supplémentaire est intégrée. Cette sortie représente une fuite et est associée à une variable booléenne. Son fonctionnement est le suivant si la variable est mise à 0 la sortie est fermée si elle est mise à 1 alors elle est ouverte.

Pour extraire les listes des alarmes correspondant à un défaut, on agit comme un opérateur du système. On démarre le système à partir de la supervision jusqu'à placer le système dans l'état de fonctionnement souhaité. Une fois le régime permanent atteint, le défaut est déclenché en mettant à un une variable booléenne correspondant au défaut au niveau de l'automate. On attend la propagation du défaut jusqu'à ce que plus aucune alarme ne soit déclenchée par le système de contrôle. Après stabilisation des alarmes, on exporte la liste d'alarmes générée par le simulateur sous forme d'un fichier Excel. On répète cette opération pour chaque défaut, et pour chaque état du système afin de créer de la variabilité dans le jeu de données.

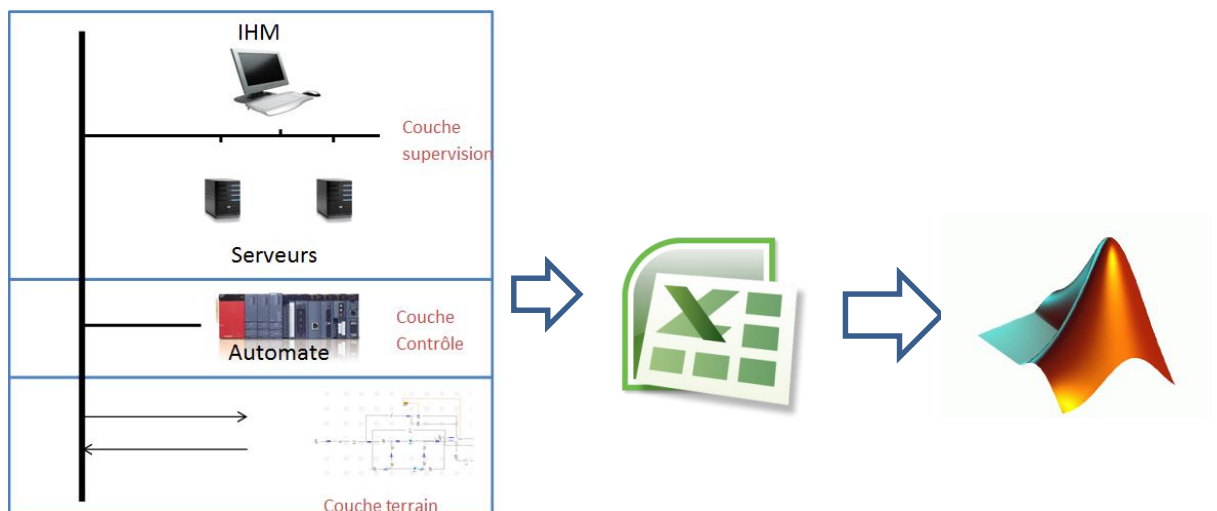


Figure 3-11 Schéma d'extraction des données du simulateur

Un total de 13 défauts a été simulé. Pour chaque défaut, six listes d'alarmes ont été générées à partir de six états de fonctionnement différents. Les critères retenus pour sélectionner les défauts après discussion avec un expert du système sont les suivants : fréquence d'apparition et difficulté de diagnostic :

Tableau 3-1 Liste des défauts simulés

Numéro	Module	Type
1	Source Gaz	Fuite dans la bouteille d'Ar
2	Source Gaz	Fuite dans la bouteille de Co2
3	Mixeur	Problème de régulation MFC ligne 1
4	Mixeur	Problème de régulation MFC de la ligne 3
5	Mixeur	Fuite vanne de sortie
6	Échappement	Problème de regulation MFC
7	Échappement	Problème de regulation MFC
8	Circuit principal	Fuite dans le Buffer
9	Pompe	Stop de la pompe 1
10	Pompe	Problème de régulation de la pompe
11	Pompe	Fuite dans la pompe
12	Distribution	Bulleur cassé ligne 2
13	Distribution	Bulleur cassé ligne 6

Tous ces défauts sont des défauts permanents et peuvent déclencher entre 20 et 150 alarmes.

3.3. Validation du simulateur

Afin de valider le modèle, une comparaison entre certaines variables physiques du système en production et ces mêmes variables calculées par le simulateur a été réalisée. Un défaut a également pu être déclenché sur le système réel et comparé avec le système modélisé. Enfin, le simulateur a été testé par deux experts du système gaz.

3.3.1. Comparaison des variables physiques

Pour opérer la comparaison entre le simulateur et le système réel, on place le simulateur dans l'état initial du système réel, avant démarrage. On démarre les deux systèmes et on compare les principales variables physiques.

Un filtre « dead-band » permet de réduire le nombre de points enregistrés par système de supervision, surtout lorsque le signal évolue peu. Cependant si la valeur de la « dead-band » est mal réglée, il arrive que l'on perde un peu d'information. Malheureusement, les données réelles ont été enregistrées avec une dead-band parfois trop élevée, d'où la différence parfois observée entre les mesures réelles et les simulées.

Les Figure 3-12 et Figure 3-13 présentent le débit sur la ligne de gaz CO₂ du mixeur et la pression du buffer qui se situe sur le circuit principal du gaz et montrent une bonne concordance entre les variables réelles et simulées.

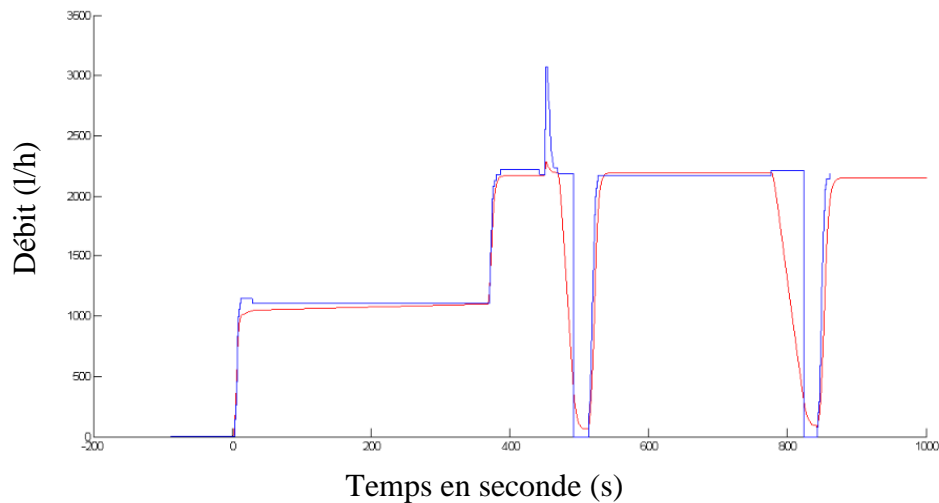


Figure 3-12 Débit sur la ligne de gaz CO₂ du Mixeur (Courbe bleue : données réelles, Courbe rouge : données simulés)

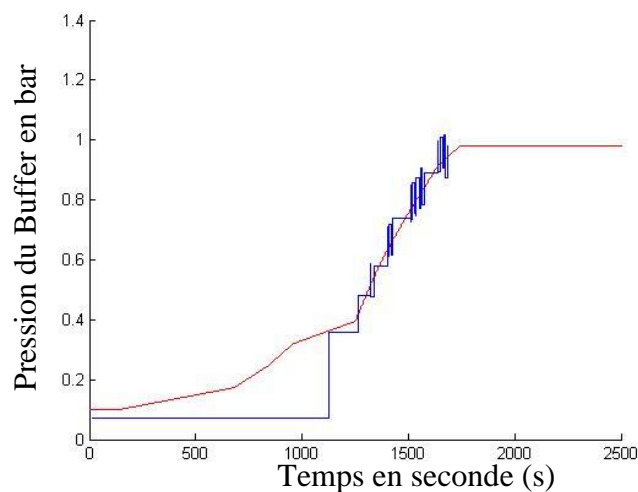


Figure 3-13 Pression dans le buffer du système gaz (Courbe bleue : données réelles, Courbe rouge : données simulés)

3.3.2. Comparaison des listes d'alarmes

Afin d'évaluer également le simulateur en situation anormale, nous avons pu déclencher un défaut sur le système réel et le comparer avec le même défaut généré dans le simulateur. Le défaut simulé est celui d'une panne moteur dans le module pompe. On présente les deux listes d'alarmes du même défaut l'une générée par le système réel et l'autre par le simulateur. Les deux listes d'alarmes sont présentées sous forme matricielle. Chaque liste est transformée en un vecteur de dimension égale au nombre d'alarmes pouvant être déclenchées par le système (au nombre de 460). Chaque ligne du vecteur représente une alarme. Si l'alarme est déclenchée lors du défaut, la ligne correspondant à l'alarme est mise à 1 dans le vecteur, sinon elle reste à zéro. On applique cette représentation aux deux listes d'alarmes. La matrice finale contenant les deux défauts est la concaténation des deux vecteurs. On représente cette matrice sous forme d'image.

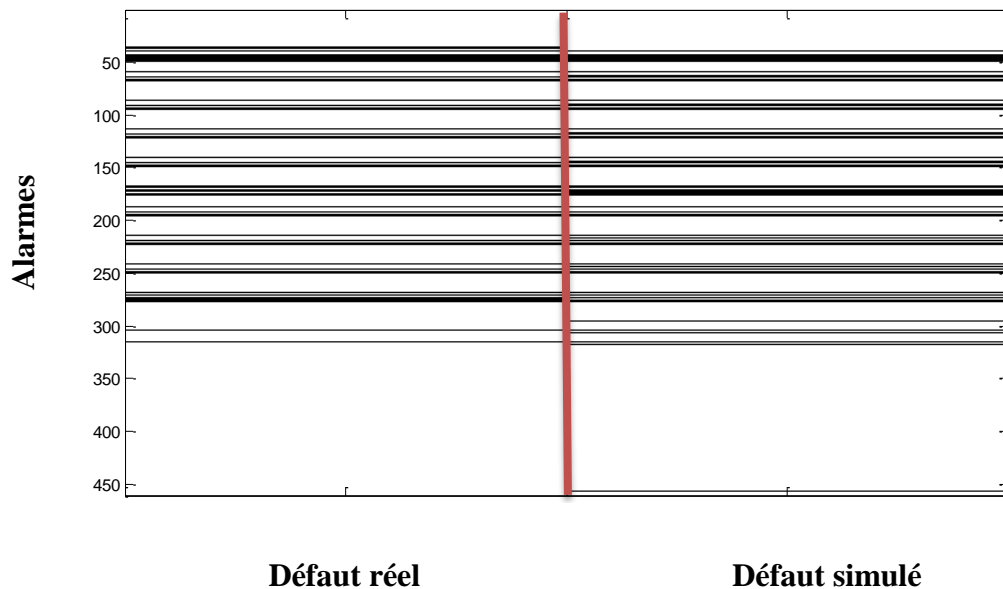


Figure 3-14 Comparaison de liste d'alarmes entre simulateur et système réel

On constate une forte similarité entre les deux listes d'alarmes avec une grande majorité d'alarmes communes aux deux listes d'alarmes.

3.3.3. Avis des experts

Le simulateur a, tout le long de sa conception, été confronté à l'œil avisé des experts et notamment à celui de l'expert à l'origine de la conception du procédé. Une fois réalisé, il a été utilisé par les experts afin d'évaluer la qualité. Plusieurs défauts ont été simulés en aveugle et les experts ont analysé les listes d'alarmes fournies et pu diagnostiquer correctement les défauts. Les retours des experts ont été très positifs. D'après eux, le simulateur correspond à ce qui serait attendu du système réel.

Voici un petit extrait d'une conversation avec le concepteur des systèmes gaz, après une évaluation du simulateur après le déclenchement d'un défaut dans le procédé (une fuite sur le bulleur) :

«... I particularly like to think back to the day when we found the “bubbler leak” in the simulator,-and the way we found it was exactly the same, that would be necessary on the real hardware»

« J'apprécie particulièrement de me remémorer le jour où nous avons trouvés la « fuite du bulleur » dans le simulateur,- et la façon dont on l'a trouvé fut exactement celle que l'on aurait utilisé sur le vrai système »

3.4. Conclusion

Ce chapitre présente le simulateur des systèmes gaz installé sur les détecteurs du LHC. Après une description du système et des différents modules, nous avons présenté le logiciel utilisé au CERN pour la simulation. Il repose sur une philosophie de développement orientée objet où chaque composant du système tel que les vannes, les pompes etc... sont modélisés séparément et finalement liés afin de créer le système complet. La bibliothèque de composants développée pour modéliser différents composants du système gaz y est explicitée et les fonctions de transfert de chaque composant sont détaillées en annexe. Elles reposent principalement sur les lois de thermodynamique et de mécanique des fluides. Le modèle du procédé a été intégré au système de contrôle afin d'avoir un simulateur complet et utilisable. Enfin, le simulateur a été validé à l'aide de plusieurs méthodes. La première repose sur la comparaison avec le système réel lors d'une phase de démarrage et montre des résultats proche du système réel. La seconde est un test en situation de défaut d'une pompe où la liste d'alarmes du système réel est comparée à celle générée par le simulateur. Enfin, le simulateur a été donné aux experts du système gaz pour validation dont le retour a été très positif.

4. Diagnostic par comparaison de vecteurs d'alarmes

Dans ce chapitre, nous détaillons la méthode de diagnostic proposée quand les listes alarmes sont représentées sous forme de vecteurs, c'est-à-dire lorsque l'ordre d'apparition des alarmes n'est pas pris en compte. L'ensemble des exemples représentant un défaut particulier est synthétisé sous un prototype de défaut. Des poids définissant la pertinence d'une alarme dans le prototype de défauts sont calculés, ce qui permet alors la définition d'une distance pondérée permettant la comparaison d'un vecteur d'alarmes généré par un défaut inconnu à un prototype de défaut.

4.1. Définition du problème/formalisme

On se propose dans ce paragraphe de mettre en place les notations et le formalisme qui seront utilisés par la suite pour décrire la méthodologie.

Soit N , le nombre d'alarmes total que le système de contrôle peut déclencher et faire apparaître sur la liste de supervision. L'ensemble des alarmes produites sur occurrence d'un défaut est représentée par un vecteur V , de taille N .

$$V^T = (v_1, \dots, v_j, \dots, v_N) \quad 4-1$$

On considère qu'une alarme a été déclenchée si elle apparaît dans la liste d'alarmes de la supervision sous l'état activé, comme défini au chapitre 1, c'est-à-dire si elle s'est déclenchée au moins une fois. L'élément j du vecteur correspond à l'alarme $n^o j$. v_j prend la valeur 0 si l'élément n'a jamais été déclenché sur l'intervalle d'analyse. Cet intervalle débute sur l'occurrence de la première alarme et se termine après un temps T correspondant à la stabilisation de la faute c'est-à-dire au moment où plus aucune alarme n'est déclenchée suite à la faute. v_j prend la valeur 1 si l'alarme a été déclenchée au moins une fois au cours de l'intervalle d'analyse. Par exemple, une alarme activée et désactivée plusieurs fois au cours de l'intervalle d'analyse prendra la valeur 1.

Afin de mieux comprendre la représentation vectorielle, prenons l'exemple d'un système pouvant générer 6 alarmes différentes. Considérons maintenant la liste d'alarmes générée lors d'une faute notée f^i :

Tableau 4-1 Exemple de liste d'alarmes

Heure de déclenchement	Nom de l'Alarme	Etat
00:00:01	A1	Activé
00:00:02	A4	Activé
00:00:02	A1	Désactivé
00:00:03	A2	Activé

Le vecteur associé à cette liste d'alarmes est noté V . Il est de taille 6:

$$V = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \begin{matrix} A1 \\ A2 \\ A3 \\ A4 \\ A5 \\ A6 \end{matrix} \quad 4-2$$

Nous disposons de k exemples représentant les m défauts que l'on souhaite diagnostiquer. Chaque exemple est représenté par le vecteur composé des alarmes déclenchées depuis l'apparition de la première alarme (synonyme d'un état anormal du système) jusqu'à l'instant de stabilisation de la faute. On appellera F l'ensemble des m défauts différents contenus dans la base d'apprentissage.

$$F = \{f^1, \dots, f^i, \dots, f^m\} \quad 4-3$$

Avec f^i le défaut numéro i .

Le vecteur d'alarmes de l'exemple $q \in \{1 \dots k\}$ du défaut $i \in \{1 \dots m\}$ est représenté par :

$$V^{i,q} = (v_1^{i,q}, \dots, v_j^{i,q}, \dots, v_N^{i,q}) \quad 4-4$$

On appellera A l'ensemble d'apprentissage, composé de vecteurs d'alarmes associées à un défaut, f^i , avec i allant de 1 à m :

$$A = \left\{ \{(V^{1,1}), \dots, (V^{1,r}), \dots, (V^{1,k})\} \dots, \{(V^{i,1}), \dots, (V^{i,r}), \dots, (V^{i,k})\} \right\} \quad 4-5$$

, ..., { (V^{m,1}), ..., (V^{m,k}) }

Où $V^{i,r}$ représente l'exemple r du défaut f^i :

La taille du jeu de données d'apprentissage est donc de $k*m$ vecteurs de taille N . Pour simplifier les expressions mathématiques, on suppose dans ce mémoire que le nombre d'exemples par défaut, k , est le même quel que soit le défaut. La méthode proposée reste cependant valide si le nombre d'exemples varie d'un défaut à l'autre.

4.2. Prototypes

4.2.1. Définition

Compte tenu de la variabilité intrinsèque au système physique (différents modes de fonctionnement, environnement extérieur, évolution des composants dans le temps, ...), un même défaut ne produit pas systématiquement les mêmes alarmes. L'ensemble des exemples d'un même défaut est synthétisé en un vecteur, appelé prototype, qui représente de manière unique le défaut. Le prototype du défaut i , noté P^i , est un vecteur binaire de taille N :

$$P^{iT} = (p_1^i, \dots, p_j^i, \dots, p_N^i)$$

Avec

$$\begin{cases} p_j^i = 1 \text{ si } fr_j^i = \frac{\sum_{l=1}^k v_j^{i,l}}{k} \geq 0.5 \\ p_j^i = 0 \text{ sinon} \end{cases} \quad 4-6$$

Le prototype est construit à partir des exemples du défaut qu'il représente. Le $j^{ème}$ élément, p_j^i , du vecteur P^i prend la valeur 1 si l'alarme j apparaît dans les différents exemples du défaut avec une fréquence supérieure ou égale à 0.5.

La valeur de seuil de 0.5 permet de ne garder que les alarmes qui se déclenchent au moins une fois sur deux, sur l'occurrence du défaut i . Les alarmes se produisant avec une fréquence d'apparition inférieure à 0.5 sont considérées comme peu représentatives du défaut. Ce choix de représentation des prototypes permet de conserver l'information sous forme binaire, ce qui rend le prototype plus facilement interprétable et permet une meilleure compréhension du lien entre alarmes et défauts. En effet, une autre approche aurait pu consister à transcrire dans le prototype la fréquence d'apparition de l'alarme et obtenir ainsi un vecteur numérique, dissimulant alors l'information binaire portée par l'alarme.

4.3. Poids

4.3.1. Définition

Les alarmes déclenchées lors d'un défaut n'ont pas toutes la même importance pour décrire ce défaut. Comme il a été dit précédemment, du fait du non déterminisme des procédés continus du CERN, les alarmes générées lors d'un même défaut ne sont pas toujours les mêmes. Par ailleurs, certaines alarmes sont déclenchées quasi systématiquement, quel que soit le défaut présent. La quantité d'informations que véhiculent ces alarmes est moins grande que celle d'alarmes qui ne sont déclenchées que lors de l'occurrence d'un défaut spécifique.

Afin de caractériser la pertinence des alarmes pour un défaut particulier, un vecteur poids, W^i , est associé à chaque prototype de défaut P^i .

$$W^{iT} = (w_1^i, \dots, w_j^i, \dots, w_N^i) \quad 4-7$$

w_j^i caractérise la pertinence de l'alarme j pour le défaut i . L'idée exprimée par w_j^i est la suivante:

- Une alarme j est d'autant plus pertinente pour le défaut i que sa valeur (0 ou 1) est identique pour un grand nombre d'exemples du défaut i .

- Une alarme j est pertinente pour le défaut i comparé à un ensemble d'autres défauts si la valeur prise par l'alarme dans le défaut i est le contraire de la valeur prise par l'alarme sur l'occurrence des autres défauts.

Chaque poids w_j^i s'exprime donc suivant deux termes, α_j^i et β_j^i , de la manière suivante, chacun quantifiant une des affirmations précédentes :

$$w_j^i = (2 \cdot \alpha_j^i - 1) \cdot (1 - \beta_j^i) \quad 4-8$$

Les termes α_j^i et β_j^i sont définis par :

$$\alpha_j^i = \frac{\sum_{q=1}^k \delta(v_j^{i,q} - p_j^i)}{k} \quad \beta_j^i = \frac{\sum_{h \in H^i} \frac{\sum_{q=1}^k \delta(v_j^{i,q} - p_j^i)}{k}}{g} \quad 4-9$$

H^i est un ensemble de g ($g \leq m-1$) défauts contenus dans la base d'apprentissage mais excluant le défaut i : $H^i = \{1, \dots, m\} \setminus \{i\}$

Avec

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases} \quad 4-10$$

Le premier terme, $(2 \cdot \alpha_j^i - 1)$ évalue la pertinence de l'alarme j au sein du défaut i . Le terme α_j^i estime la probabilité que l'alarme j prenne la valeur p_j^i dans les exemples du défaut i . Sa valeur est comprise entre 0.5 et 1. Que p_j^i vaille 0 ou 1, le terme α_j^i est maximal et égal à 1 si l'ensemble des exemples du défaut prennent la même valeur pour l'alarme j . Le poids du terme $(2 \cdot \alpha_j^i - 1)$ est alors égal à 1. Si l'alarme ne fait pas l'unanimité, α_j^i est alors compris entre 0.5 et 1. Plus la valeur de α_j^i est proche de 0.5, plus la valeur $(2 \cdot \alpha_j^i - 1)$ est faible et proche de 0. Le terme $(2 \cdot \alpha_j^i - 1)$ attribue un faible poids aux alarmes dont la probabilité d'apparition est proche de l'aléatoire pur (valeur de 0.5). Ces alarmes sont considérées comme peu porteuses d'information. Dans le cas d'une alarme purement aléatoire, α_j^i vaut 0.5, alors $(2 \cdot \alpha_j^i - 1)$ vaut 0. Le terme $(2 \cdot \alpha_j^i - 1)$ permet de donner plus d'importance aux alarmes caractéristiques d'un défaut (apparaissant souvent pour les exemples d'un même défaut) et à l'inverse filtrer les alarmes à caractère aléatoire, peu porteuses d'information.

Le deuxième terme $(1 - \beta_j^i)$ évalue la pertinence de l'alarme j par comparaison avec les autres défauts présents dans l'ensemble H^i . β_j^i estime la probabilité que l'alarme i prenne la valeur p_j^i sur l'occurrence des défauts contenus dans l'ensemble H^i . Un β_j^i fort signifie que la valeur p_j^i prise par l'alarme j dans le défaut i est identique à celle prise par l'alarme j dans les vecteurs d'alarmes des autres défauts. L'alarme j est donc faiblement caractéristique du

défaut i , soit parce qu'elle est déclenchée quel que soit le défaut, (dans ce cas-là, il s'agira de bruit), soit car elle n'est jamais déclenchée quel que soit le défaut (dans ce cas-là, il s'agira d'une alarme muette). Au contraire, un β_j^i faible signifie que l'alarme j apparaît avec sa valeur contraire dans les vecteurs d'alarmes des autres défauts et qu'elle est donc plus spécifique au défaut i . Le terme $(1 - \beta_j^i)$ varie entre 0 et 1.

Les figures suivantes présentent l'évolution des différentes valeurs de w_j^i en fonction des termes α_j^i et β_j^i . La Figure 4-1 est une représentation 3D de la fonction de calcul de poids en fonction de α_j^i et β_j^i et la Figure 4-2 est une représentation 2D :

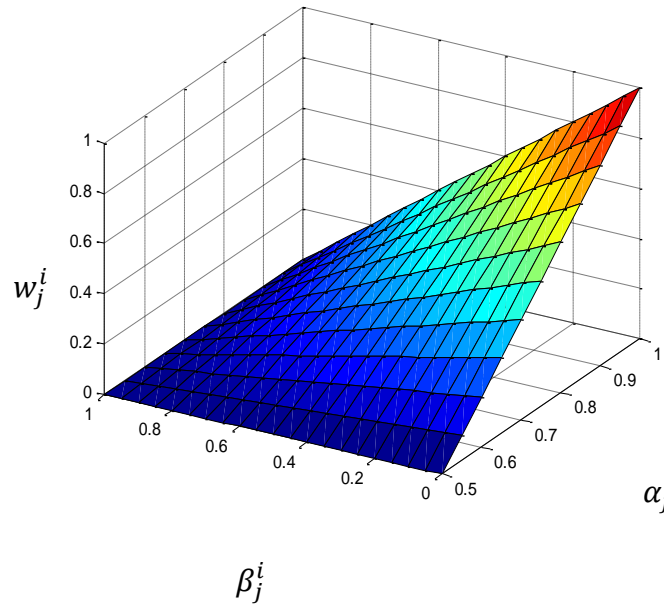


Figure 4-1 Représentation 3D de w_j^i en fonction de α_j^i et β_j^i

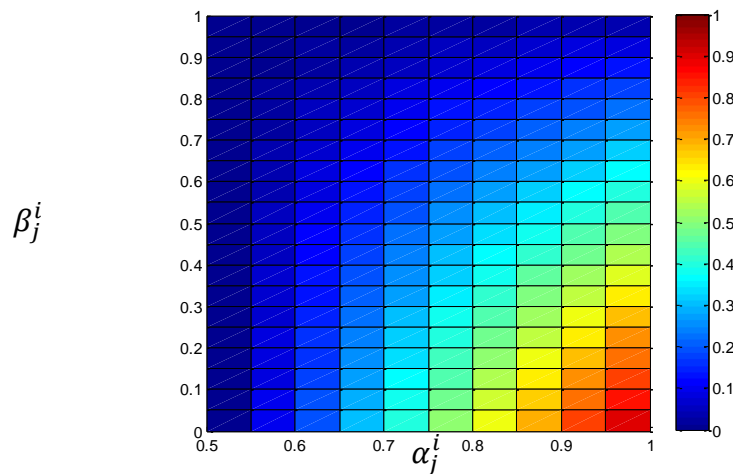


Figure 4-2 Représentation 2D de w_j^i en fonction de α_j^i et β_j^i

Les zones bleues représentent les valeurs les plus faibles de w_j^i , les rouges représentent les valeurs les plus hautes de w_j^i . Les valeurs les plus grandes sont effectivement obtenues pour un α_j^i proche de 1 et un β_j^i proche de 0. Les w_j^i faible sont obtenues pour α_j^i faible c'est-à-dire pour une probabilité d'apparition de l'alarme proche de 0.5 dans les exemples du défaut ou pour un β_j^i faible c'est-à-dire une probabilité élevée d'apparition (ou d'absence) de l'alarme dans les autres défauts. La fonction de poids définie représente donc bien le mécanisme qui consiste à donner plus de poids aux alarmes apparaissant fréquemment dans les exemples d'un même défaut et rarement dans les exemples d'autres défauts.

4.3.2. Analyse des poids

Le poids d'une alarme j associée à un défaut i est une grandeur qui varie entre 0 et 1. Ceci permet d'interpréter le poids comme une pertinence. Plus le poids est proche de 1, plus l'alarme est significative pour décrire le défaut i , par comparaison avec un ensemble d'autres défauts. A l'inverse, un poids égal à 0 met en évidence une alarme n'apportant pas d'information sur le défaut i . Ces poids sont donc directement interprétables par l'opérateur. La décomposition des poids en deux termes, $(2\alpha_j^i - 1)$ et $(1 - \beta_j^i)$, qui varient tous deux entre 0 et 1, permet de mieux spécifier la pertinence de l'alarme. Celle-ci peut être non pertinente car apparaissant de manière aléatoire sur l'occurrence du défaut i ou car se déclenchant fréquemment sur l'occurrence des autres défauts. La pondération que les deux termes apportent à chaque alarme est équivalente. Le fait qu'une alarme soit représentée par une valeur identique au sein des exemples du défaut i a le même impact que le fait que l'alarme soit représentée par sa valeur contraire dans la majorité des exemples des autres défauts.

Outre le fait qu'il soit directement interprétable, la variation de chaque poids entre 0 et 1 permet la comparaison des alarmes les unes par rapport aux autres. L'ensemble de la base d'apprentissage peut être synthétisée en une matrice de poids, où chaque colonne i représente le vecteur des poids associé à un défaut i et chaque ligne j le poids d'une alarme j pour chaque défaut. La matrice des poids peut être présentée à l'opérateur sous forme d'image, ce qui permet une interprétation visuelle de l'ensemble de la base d'apprentissage. Pour une analyse plus fine, la matrice des poids peut être décomposée en deux sous matrices, la matrice des poids $(2\alpha_j^i - 1)$ et la matrice des poids $(1 - \beta_j^i)$. Une illustration de ces matrices sera présentée au chapitre 6 présentant les résultats obtenus par la méthode.

Une alarme peut être comparée aux autres alarmes au sein du défaut i par l'analyse du vecteur W^i . L'analyse de la matrice des poids est faite par colonne. Les alarmes portant les plus grand poids sont les alarmes les plus discriminantes pour le défaut i . Si l'ensemble des poids a une valeur proche de zéro, cela informe sur le fait qu'aucune des alarmes générées par le système de contrôle ne permet d'isoler le défaut parmi l'ensemble des autres défauts. Celui-ci n'est pas diagnostiquable à partir de la liste d'alarmes sélectionnées. En effet, soit les alarmes n'apparaissent qu'une fois sur deux lors de l'occurrence du défaut, soit les alarmes se

produisent fréquemment, elles se produisent donc sur l'occurrence de tous les autres défauts de la base. L'opérateur peut en conclure qu'il faut modifier le système de contrôle en ajoutant de nouvelles alarmes. Une autre conclusion peut être que la représentation vectorielle choisie n'est pas adaptée car la simple analyse des alarmes présentes ou absentes ne suffit pas et que l'ordre dans lequel elles apparaissent doit être pris en compte.

Une alarme j peut aussi être interprétée au sein de la base d'apprentissage, en analysant la valeur prise par le poids de l'alarme pour chaque défaut dans la base. L'analyse de la matrice des poids est alors effectuée en ligne. Les valeurs importantes observées sur une ligne mettent en évidence les défauts bien spécifiés par l'alarme. Une ligne dont les poids prennent des valeurs globalement faibles traduit une alarme peu discriminante pour l'ensemble des défauts que l'on cherche à isoler. Cette analyse pourrait permettre le « nettoyage » de la base des alarmes pouvant être déclenchées. Ainsi, les alarmes associées à un β_j^i élevée sont des alarmes qui sont déclenchées la majorité du temps avec une entropie faible. Elles sont donc peu utiles au diagnostic et pourraient être supprimées de la base afin d'alléger le nombre de données présentées à l'opérateur.

L'analyse faite à partir des poids peut être différente en fonction de la liste des défauts utilisés pour les calculer. En effet, notons que le fait de changer la liste des défauts présents dans H^i modifie la valeur de $(1 - \beta_j^i)$. Cela permet d'adapter la notion de « poids caractérisant une alarme j pour un défaut i par comparaison à un ensemble de défauts » à différents contextes. On peut caractériser le défaut i par rapport à l'ensemble des autres défauts de la base d'apprentissage. Les poids sont alors calculés en utilisant une stratégie de « seul contre tous ». Dans ce cas $H^i = \{1, \dots, m\} \setminus \{i\}$. Cette analyse met en évidence les alarmes se déclenchant fréquemment, et donc polluant l'information provenant du système de contrôle.

On peut aussi caractériser le défaut i par rapport à un nombre de défauts réduit, en sélectionnant par exemple les défauts ayant tendance à déclencher les mêmes alarmes. Dans ce cas, le nombre g de défauts contenus dans H^i est plus faible. Il est plus facile d'atteindre une majorité forte, voire l'unanimité, ce qui a pour conséquence d'attribuer à $(1 - \beta_j^i)$ des valeurs plus extrêmes (0 ou 1). Le poids des alarmes discriminants les défauts proches est renforcé. En effet, la valeur β_j^i étant fortement dépendante de l'ensemble H^i , un nombre important d'éléments dans H^i a tendance à moyenner les valeurs de β_j^i .

La démarche extrême consiste à comparer le défaut i à un autre défaut l . Dans ce cas, $H^i = \{l\}$. La stratégie utilisée est alors une stratégie de « l'un contre l'autre ». Les poids permettent alors de clairement mettre en évidence les alarmes qui différencient les 2 défauts. Un poids fort ne peut être atteint que si α_j^i et $(1 - \beta_j^i)$ prennent tous les deux une valeur élevée, caractérisant une alarme j se produisant très fréquemment sur le défaut i et très rarement sur le défaut l . Afin de différencier deux défauts (i et l) entre eux et calculer les poids avec une base

réduite à deux défauts, on propose de modifier le calcul des poids en utilisant la formule de calcul de poids suivante :

$$w_j^{i,l} = \max((2\alpha_j^i - 1) \cdot (1 - \beta_j^i), (2\alpha_j^l - 1) \cdot (1 - \beta_j^l)) \quad 4-11$$

Exprimer ainsi les poids des alarmes présente l'avantage d'avoir une valeur symétrique et d'obtenir ainsi un unique vecteur de poids entre les deux défauts. Cette démarche prend tout son intérêt quand deux défauts vont déclencher un grand nombre d'alarmes communes. L'analyse automatique des poids va permettre d'isoler les rares alarmes discriminant les deux défauts.

4.4. Définition de la distance pondérée D_{vp}

Afin de comparer le vecteur engendré par le défaut inconnu que l'on souhaite diagnostiquer, noté V , avec les prototypes de défauts, on définit une distance de la manière suivante :

$$D_{vp}(V, P^i) = \frac{W^{iT} |V - P^i|}{\sum_{j=1}^N w_j^i} = \frac{\sum_{j=1}^N w_j^i |v_j - P_j^i|}{\sum_{j=1}^N w_j^i} \quad 4-12$$

La distance définie est une distance vectorielle pondérée et normée. Celle-ci varie entre 0 et 1. Le vecteur de pondération W^i est spécifique à chaque prototype. Le fait de normer la distance permet une comparaison du vecteur avec les différents prototypes de défauts, même si les vecteurs W^i sont dépendants du prototype. Normer la distance est indispensable pour que celle-ci ait un sens. Sans cette opération, la distance entre un vecteur et un prototype dont l'ensemble des poids serait de faible valeur serait petite sans que l'on puisse en conclure que le vecteur et le prototype soient semblables.

Normer la distance par la somme des poids va permettre de renforcer l'impact des alarmes discriminantes pour le défaut, car leur poids relatif sera augmenté. Supposons que le défaut i soit caractérisé par une unique alarme parmi l'ensemble des N alarmes possibles. Son poids prend la valeur 1 alors que ceux des autres sont proches de 0. La distance est alors calculée principalement à partir de cette unique alarme car son poids relatif dans la distance est égal à 1. A l'inverse, si toutes les alarmes sont discriminantes et portent un poids de 1, chacune interviendra de manière équivalente (poids de $1/N$) dans la distance, qui sera ainsi implicitement ramenée à une distance de Hamming (équation 6-1).

Ceci justifie aussi l'intérêt d'affecter un poids nul à des alarmes se déclenchant sur l'occurrence de tous les défauts, et donc de l'intérêt du terme $(1-\beta^i)$ dans le calcul de distance. En effet, sur une alarme toujours déclenchée, la différence entre la valeur de l'alarme dans le vecteur à diagnostiquer et le prototype est dans ce cas toujours 0. L'alarme n'a donc pas d'impact dans le calcul de la distance, et ceci quel que soit le poids qui lui est associé. Cependant, le poids nul attribué à ces alarmes a pour effet de renforcer les poids

relatifs attribués aux alarmes discriminantes dans le calcul de la distance normée par la somme des poids.

Notons cependant que le cas particulier où tous les poids sont nuls pose un problème mathématique de division par 0. Ce cas peut cependant être traité à l'avance, par l'analyse des poids associés à chaque prototype, comme nous l'avons spécifié dans le paragraphe 4.3.2.

Notons également que la distance pondérée proposée mesure la dissimilarité entre un vecteur et un prototype. Plus elle est grande, moins le vecteur et le prototype se ressemblent. À l'inverse, on peut définir une mesure de similarité, variant entre 0 et 1, de la manière suivante :

$$Sim(V, P^i) = 1 - D(V, P^i) \quad 4-13$$

On a ici, comme pour la distance entre les vecteurs et les prototypes, une mesure variant entre 0 et 1. Mais à l'inverse on mesure une similarité entre les vecteurs. Cette mesure de similarité peut être utilisée comme une mesure de confiance dans le diagnostic. Une valeur de 0 voulant dire que l'on est sûr que les deux vecteurs n'appartiennent pas au même défaut et une valeur de 1 signifie que l'opérateur peut avoir confiance dans le diagnostic. Elle fournit donc une information supplémentaire lorsque le diagnostic est rendu à l'opérateur en ajoutant un degré de confiance.

4.5. Méthodes de décisions :

Nous proposons dans ce paragraphe plusieurs méthodologies de décisions basées sur les distances entre le vecteur à diagnostiquer et les prototypes contenus dans la base d'apprentissage. Deux problèmes sont à résoudre lors de l'analyse d'un nouveau vecteur d'alarmes :

- Le vecteur est-il généré par un défaut présent dans la base d'apprentissage ou s'agit-il d'un défaut encore inconnu ?
- Si il s'agit d'un défaut présent dans la base, de quel défaut s'agit-il ?

4.5.1. Rejet d'un vecteur d'alarmes :

Avant d'effectuer un diagnostic et d'associer un défaut au vecteur d'alarmes à diagnostiquer, il est nécessaire de vérifier que ce vecteur est bien généré par un défaut présent dans l'ensemble d'apprentissage. Pour ce faire, on procède à un test en rejet. On considère que si la plus petite distance entre les prototypes de la base et le vecteur représentant le défaut inconnu est supérieure à un seuil de rejet noté T_r , alors le défaut est considéré comme inconnu et n'appartient pas à l'ensemble F des défauts inclus dans la base. Lors du calcul de distance en vue d'un rejet, les poids des alarmes sont calculés en n'utilisant que la part des poids associé à α_j^i . Les poids β_j^i sont mis systématiquement à 1.

$$w_j^i = (2\alpha_j^i - 1) \quad 4-14$$

La partie $(1-\beta_j^i)$ est utilisée pour accentuer le poids des alarmes qui discriminent un défaut par rapport à un ensemble d'autres défauts. Au moment du test en rejet, il n'y a pas de certitude que le défaut à diagnostiquer soit présent dans la liste de défauts connus. Se focaliser sur les alarmes qui discriminent ces défauts entre eux est dangereux à ce stade. En effet, supposons que certaines alarmes soient systématiquement présentes (valeur à 1) quels que soient les défauts de la base d'apprentissage. L'utilisation du terme $(1-\beta_j^i)$ annule alors le poids de ces alarmes pour tous les prototypes de défauts. Supposons que ces mêmes alarmes n'apparaissent pas (valeur à 0) dans le vecteur à diagnostiquer. Ceci est une indication forte que le défaut inconnu est différent des défauts contenus dans la base de connaissance. Or, les poids associés valant 0, cette information ne sera pas prise en compte dans le calcul de la distance.

En ne gardant que la partie $w_j^i = (2\alpha_j^i - 1)$ des poids, on ne compare le vecteur généré par le défaut inconnu et le prototype de défaut P^i que sur les alarmes dont l'apparition n'est pas complètement aléatoire sur l'occurrence du défaut i . En effet, dans le cas d'alarmes où le poids $w_j^i = (2\alpha_j^i - 1)$ associé est proche de 0, le fait que l'alarme soit présente ou non dans le vecteur de défaut inconnu n'est pas informatif pour reconnaître le défaut. La distance pondérée utilisée :

$$Dvp(V, P^i) = \frac{\sum_{j=1}^N w_j^i |v_j - p_j^i|}{\sum_{j=1}^N w_j^i}$$

Avec

$$w_j^i = (2\alpha_j^i - 1)$$

renforce l'importance des alarmes à caractère déterministe pour le défaut i .

L'algorithme de diagnostic proposé est donc le suivant, il se décompose en deux parties, une première hors ligne pour créer les prototypes à partir d'une base d'apprentissage et une seconde en ligne pour le test en rejet d'un vecteur inconnu, V :

Partie 1: Création des prototypes

Pour i allant de 1 à m
 Pour j allant de 1 à N

$$fr_j^i = \frac{\sum_{l=1}^k v_j^{i,l}}{k}$$
 si $fr_j^i \geq 0.5$

$$p_j^i = 1$$
 Sinon

$$p_j^i = 0$$
 fin si

$$\alpha_j^i = \frac{\sum_{q=1}^k \delta(v_j^{i,q} - p_j^i)}{k}$$

$$w_j^i = (2\alpha_j^i - 1)$$
 Fin pour

$$P^i = (p_1^i, \dots, p_j^i, \dots, p_N^i)$$

$$W^i = (w_1^i, \dots, w_j^i, \dots, w_N^i)$$
 Fin Pour

Partie 2 test en rejet :

Soit V le vecteur représentant le défaut à diagnostiquer :

Pour i allant de 1 à m

$$d^i = D_{vp}(V, P^i) = \frac{\sum_{j=1}^N w_j^i |v_j - p_j^i|}{\sum_{j=1}^N w_j^i}$$

Fin pour

Si $\exists i d^i \leq T_r$ alors le défaut est contenu dans la base d'apprentissage
 Sinon le défaut n'est pas contenu dans la base d'apprentissage

4.5.2. Affectation d'un vecteur d'alarmes à un défaut contenu dans la base d'apprentissage

Une fois le vecteur testé en rejet et considéré comme connu, celui-ci peut être affecté à un défaut f^i contenu dans l'ensemble F . Les valeurs β_j^i peuvent être utilisées pour faciliter la discrimination d'un défaut par rapport à un autre. Les poids utilisés pour chaque prototype sont donc : $w_j^i = (2\alpha_j^i - 1)(1 - \beta_j^i)$ et la distance utilisée est D_{vp} .

Plusieurs stratégies peuvent être appliquées pour assister l'opérateur dans sa prise de décision.

4.5.2.1. Méthode du plus proche voisin:

La première méthode propose d'associer le défaut inconnu avec le prototype dont la distance est la plus petite. Cette méthode est aussi connue sous le nom de l'algorithme du k-plus proche voisin (k ppV), avec $k = 1$ dans notre cas.

Elle compare la liste d'alarmes à diagnostiquer avec les différents prototypes et associe le défaut inconnu avec le prototype ayant la distance minimum. Une telle méthode peut s'exprimer de façon mathématique de la façon qui suit :

$$\text{si } D_{vp}(V, P^i) = \min_{m \in \{1, \dots, f\}} D_{vp}(V, P^m) \text{ alors } S = \{f^i\} \quad 4-15$$

Avec S l'ensemble des solutions

Les poids β_j^i sont calculés dans ce cas en prenant en compte l'ensemble des défauts de la base d'apprentissage. La stratégie de calcul des poids est celle du « seul contre tous ».

On peut décrire l'algorithme en pseudo code. On note $\Delta = (d^1, \dots, d^m)$ l'ensemble des mesures de distances entre le vecteur de défaut inconnu V et chacun des m prototypes. Il est aussi exécuté en deux temps, dans un premier temps hors ligne et un autre en ligne :

1^{er} partie en ligne : Elle est similaire à celle du test en rejet, seuls les poids sont calculés de manière différente en prenant en compte β_j^i

Pour i allant de 1 à m

Pour j allant de 1 à N

$$fr_j^i = \frac{\sum_{l=1}^k v_j^{i,l}}{k}$$

$$\text{si } fr_j^i \geq 0.5$$

$$p_j^i = 1$$

$$\text{Sinon } fr_j^i < 0.5$$

$$p_j^i = 0$$

$$\text{fin si}$$

$$\alpha_j^i = \frac{\sum_{q=1}^k \delta(v_j^{i,q} - p_j^i)}{k} \quad \text{et} \quad \beta_j^i = \frac{\sum_{h \in H^i} \frac{\sum_{q=1}^k \delta(v_j^{i,q} - p_j^i)}{k}}{g}$$

$$w_j^i = (2\alpha_j^i - 1)(1 - \beta_j^i)$$

Fin pour

$$P^{iT} = (p_1^i, \dots, p_j^i, \dots, p_N^i)$$

$$W^{iT} = (w_1^i, \dots, w_j^i, \dots, w_N^i)$$

Fin Pour

2ème partie en ligne :

$$\Delta = \emptyset$$

Pour i allant de 1 à p

$$d^i = D_{vp}(V, P^i) = \frac{\sum_{j=1}^N |v_j - P_j^i|}{\sum_{j=1}^N w_j^i}$$

$$\Delta = \Delta \cup \{d^i\}$$

Fin Pour

$$\text{Trouver } i \text{ tel que } d^i = \min_{m \in \{1, \dots, f\}} D_{vp}(V, P^m)$$

$$S = \{f^i\}$$

Cette méthode présente l'avantage de ne présenter qu'un unique diagnostic à l'opérateur. De plus, elle ne nécessite pas de paramètres de réglage.

4.5.2.2. Méthode par seuillage :

Cette méthode consiste à fixer un seuil de détection T_d , compris entre 0 et 1, sur la distance du vecteur aux prototypes et de sélectionner l'ensemble des défauts dont la distance aux prototypes est inférieure à T_d . Les poids β_j^i sont là encore calculés dans une stratégie du « seul contre tous ».

Considérons l'ensemble Δ des mesures de distances entre le défaut inconnu et les différents prototypes calculés à l'aide de la distance D_{vp} .

Notons d^i la mesure de distance entre le défaut inconnu et le prototype P^i

On définit donc l'ensemble S des solutions possibles par l'algorithme :

$$S = \{ F \mid d^i \leq T_d \} \quad 4-16$$

On peut décrire l'algorithme en pseudo code en notant V la représentation vectorielle du défaut à diagnostiquer et Δ l'ensemble des distances entre le défaut inconnu et chacun des prototypes :

```

S = ∅

Pour i allant de 1 à N
    di = Dvp(V, Pi)
    Si di ≤ Td
        Alors
            S = S ∪ {fi}
        sinon
            Fin Si
    Fin Pour

Retourner S
    
```

La méthode de seuillage peut donner un ensemble de solutions compris entre l'ensemble vide et la totalité des défauts contenus dans la base d'apprentissage c'est-à-dire entre 0 et m solutions. Le cas où aucune distance n'est inférieure au seuil de détection T_d (et donc que l'ensemble de solutions est l'ensemble vide) correspond à une seconde étape de rejet. Si le cardinal de l'ensemble des solutions est strictement supérieur à 1, cela signifie qu'il existe plusieurs défauts susceptibles de correspondre et que l'algorithme n'est pas en

mesure de fournir un diagnostic unique, du fait de la proximité de plusieurs défauts avec le défaut à diagnostiquer. On peut imaginer que l'algorithme propose alors une liste de défauts possibles. Le seuil T_d est un paramètre à régler. Il faut le choisir de manière à s'assurer que le défaut réellement associé au vecteur (nommé bon diagnostic) soit présent dans l'ensemble des solutions proposées, sans toutefois proposer une liste trop longue de solutions.

La deuxième étape de rejet, réalisée dans cette étape de l'algorithme, est complémentaire du rejet effectué en première étape. En effet, lors de la seconde étape, l'algorithme se concentre sur les alarmes spécifiques à chaque défaut, en ne tenant pas compte des alarmes non spécifiques grâce au terme $(1 - \beta_j^i)$. Si, lors de la deuxième étape, le vecteur est rejeté comme défaut inconnu, cela signifie que les alarmes spécifiques aux défauts contenus dans la base prennent des valeurs contraires à celles des prototypes. Le vecteur a pu passer la première étape de rejet car le vecteur provenant du défaut inconnu peut posséder de nombreuses alarmes en commun avec au moins un des défauts de la base. Cependant, il peut s'agir d'alarmes qui sont communes à de nombreux défauts. Illustrons les deux étapes de rejet à partir de l'exemple suivant :

Soit une base contenant $m=2$ défauts et $k=2$ exemples par défaut tel que :

$$V^{1,1} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \text{ et } V^{1,2} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

$$V^{2,1} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \text{ et } V^{2,2} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

D'où les prototypes suivant :

$$P^1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \text{ et } P^2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Soit un défaut inconnu V tel que V :

$$V = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

1^{er} étape de rejet :

On calcule les poids selon la formule $(2 \cdot \alpha_j^i - 1)$:

$$W^1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \text{ et } W^2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

On a alors

$$\begin{aligned} d^1 &= D_{vp}(V, P^1) = 0.2 \\ d^2 &= D_{vp}(V, P^1) = 0.2 \end{aligned}$$

2^{eme} étape de rejet :

On calcule les poids selon la formule $(2 \cdot \alpha_j^i - 1) \cdot (1 - \beta_j^i)$:

$$W_{2ndetape}^1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \text{ et } W_{2ndetape}^2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

On a alors

$$\begin{aligned} d_{2ndetape}^1 &= D_{vp}(V, P^1) = 0.5 \\ d_{2ndetape}^2 &= D_{vp}(V, P^2) = 0.5 \end{aligned}$$

Cet exemple illustre le cas d'un vecteur non rejeté en première étape et rejeté en deuxième. Il possède de nombreuses alarmes en commun avec les prototypes proposés (alarmes de 1 à 4). Seules les alarmes 5 et 6 diffèrent de l'un et de l'autre défaut, d'où une distance aux prototypes relativement faible (égale à 0.2) dans la première étape. Cependant, les alarmes 1 à 4 sont non spécifiques aux deux défauts de la base, elles apparaissent aussi bien dans l'un que dans l'autre. En deuxième étape, les poids des alarmes 1 à 4 sont mis à 0. Seules comptent les deux dernières alarmes dans la prise en compte du calcul de distance. La distance entre vecteur et prototype est ainsi doublée. La première étape permet donc de rejeter un vecteur associé à un défaut qui ne présente pas de réelles similarités avec les défauts de la base en s'affranchissant du problème des poids nuls apportés par le calcul des poids avec le

β_j^i . Cette première étape franchie, l'algorithme regarde plus en détail les distances entre les différents défauts de la base afin de procéder à un nouveau rejet. Chaque phase de rejet permet donc d'observer un aspect différent du défaut inconnu.

4.5.2.3. Méthode en deux étapes

Comme nous l'avons précisé précédemment au paragraphe 4.3.2, le choix de l'ensemble des défauts à partir desquels calculer β_j^i a une incidence sur la valeur des poids. Nous n'avons utilisé jusqu'à maintenant que la stratégie de calcul de poids en « seul contre tous ». Nous proposons maintenant une méthode en deux étapes où les grandeurs β_j^i , et donc les poids, sont recalculés après une première sélection de défauts possibles.

Deux stratégies sont possibles :

Stratégie 1 : Une première sélection de défauts possibles est réalisée à l'aide de la méthode proposée au paragraphe précédent, en fixant un seuil de détection sur la distance entre le vecteur et les prototypes, à partir de poids calculés avec la stratégie « seul contre tous ».

Les poids sont ensuite recalculés en utilisant l'ensemble restreint des défauts possibles puis les distances entre le vecteur et les défauts possibles sont calculées. Le défaut dont le prototype est le plus proche au sens de la nouvelle distance est retenu comme diagnostic. Cette stratégie nécessite l'utilisation d'un seuil de détection, comme la méthode par seuillage du paragraphe 4.5.2.2.

Stratégie 2 : Une deuxième stratégie repose sur la recherche des plus proches voisins, qui présente l'avantage de ne pas avoir de seuils à régler. La première étape consiste à rechercher les deux prototypes de défauts les plus proches du vecteur représentant le défaut à diagnostiquer. Les poids utilisés dans cette première étape sont ceux calculés par la stratégie du « seul contre tous ». Une fois les deux plus proches voisins sélectionnés, les poids sont recalculés en utilisant la stratégie de « l'un contre l'autre », présentée au paragraphe 4.3.2. La valeur des poids des alarmes discriminant les deux défauts est ainsi rehaussée. La distance entre le vecteur du défaut à diagnostiquer et les deux prototypes est recalculée à l'aide des nouveaux poids et le défaut dont le prototype est le plus proche au sens de cette distance est retenu comme diagnostic.

Cette méthode en deux étapes permet d'affiner la recherche en augmentant la différenciation entre les prototypes. Il s'agit d'une sorte de zoom sur les deux défauts restant.

4.6. Conclusion

Dans ce chapitre, nous avons présenté les méthodes de diagnostic utilisées quand les défauts sont représentés par des vecteurs d'alarmes. Ces méthodes se basent sur une distance pondérée et normée, calculée à partir de prototypes et de vecteurs de poids associés à chaque

défaut. Le calcul des poids permet d'évaluer l'importance de l'alarme pour chaque défaut (ou son absence). L'idée de base de calcul du poids d'une alarme est la suivante : Une alarme est d'autant plus pertinente pour un défaut que sa valeur (0 ou 1) est identique pour un grand nombre d'exemples du défaut. Une alarme est pertinente pour le défaut comparé à un ensemble d'autres défauts si la valeur prise par l'alarme dans le défaut est le contraire de la valeur prise par l'alarme sur l'occurrence des autres défauts. Les poids et les distances sont alors utilisés dans un but de diagnostic.

Une première étape où la distance du vecteur du défaut à diagnostiquer aux prototypes est seuillée permet de rejeter le défaut comme inconnu dans la base d'apprentissage ou présent dans la base d'apprentissage.

Ensuite, deux approches peuvent être adoptées pour aboutir à un diagnostic : la méthode du plus proche voisin, où le défaut dont le prototype est le plus proche du vecteur est sélectionné comme diagnostic, ou la méthode par seuillage, où une liste de défauts dont la distance est inférieure à un seuil de détection est proposée comme diagnostics possibles.

La méthode du plus proche voisin permet de ne proposer qu'une unique solution. Cependant, dans le cas de défauts se ressemblant, c'est à dire générant sensiblement les mêmes alarmes, il se peut que le bon diagnostic ne soit pas proposé comme solution. La méthode par seuillage permet, elle, de proposer plusieurs solutions, avec si, nécessaire un degré de confiance qui pourrait être calculé à partir de la distance ($1 - D_{vp}$, par exemple). La tâche de diagnostic n'est pas complètement résolue mais elle est néanmoins simplifiée pour l'opérateur qui doit se décider entre un nombre de défauts plus restreints.

L'intérêt des poids proposés dans la méthode est la possibilité de les recalculer après avoir réduit la liste de défauts possibles. C'est l'idée des méthodes en deux étapes où une première sélection de défauts possibles est faite en utilisant les poids calculés à partir de l'ensemble des défauts de la base d'apprentissage. La sélection est ensuite raffinée en recalculant les poids sur un ensemble réduit de défaut, ce qui permet de renforcer le poids des alarmes spécifiques à ces défauts.

Finalement, la définition mathématique des poids w_j^i , proposée dans ce chapitre, permet d'obtenir une information variant entre 0 et 1, comparable d'une alarme et d'un défaut à l'autre et directement interprétable pour l'opérateur. Cela permet d'extraire automatiquement de la connaissance du processus et de synthétiser cette information sous la forme d'une image. Cela va permettre à l'opérateur d'avoir une vision claire du procédé et de mieux comprendre le lien entre alarmes déclenchées et défauts.

5. Outils d'analyses de performances

Dans ce chapitre, nous introduisons les outils que nous utiliserons par la suite afin d'évaluer les performances des différentes méthodes de diagnostics proposées dans cette thèse. Après avoir présenté le découpage de la base de données en base d'apprentissage et base de validation, nous décrirons plus en détail les différents outils de mesures utilisés comme la courbe COR et les matrices de confusion.

5.1. Validation des algorithmes

5.1.1. Base d'apprentissage/ Base de validation

On considère que l'on dispose d'un jeu de données représentant m défauts avec k listes d'alarmes par défaut. La base de données contenant la totalité des listes d'alarmes sera notée B et sera donc constituée de $n = m * k$ listes d'alarmes. À partir de ces données nous allons créer deux bases distinctes la «Base d'apprentissage» (B_a) et la «Base de validation» (B_v) définies comme suit:

- La « Base d'apprentissage » contient les données des listes d'alarmes qui serviront de données d'apprentissage. Chaque liste d'alarmes de cette base sera considérée comme connue et correctement étiquetée.
- La « Base de validation » contient des listes d'alarmes des défauts considérés comme inconnus et donc à diagnostiquer par les algorithmes de diagnostic. La comparaison entre le diagnostic proposé par l'algorithme et le défaut associé à la liste d'alarmes permet l'évaluation de l'algorithme.

Lors de la création de ces bases, on prendra soin d'avoir des exemples de chaque défaut dans chacune d'elle, et de n'avoir aucune liste d'alarmes communes aux deux bases afin d'éviter tout biais lors de la validation des algorithmes. On a les relations suivantes sur les bases:

- $B_a \subset B$
- $B_v \subset B$
- $B_a \cap B_v = \emptyset$
- $B_a \cup B_v = B$

En résumé à partir des $k * m$ listes d'alarmes on peut construire deux bases bien distinctes, une base de cas de taille $(k - \mu) * m$ utilisée comme base d'apprentissage et une base de validation de taille $\mu * m$ avec $\mu \in \mathbb{N}$ et $\mu < k$.

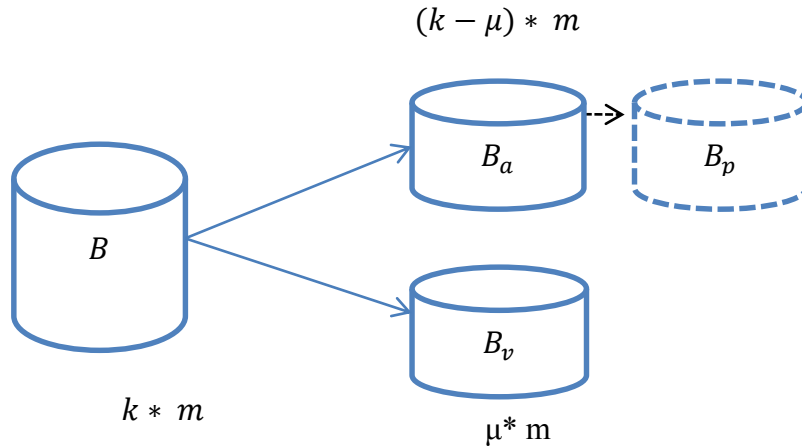


Figure 5-1 illustration de la création de B_a et B_v

Notons que les méthodes présentées dans ce travail de thèse nécessitent la création d'un prototype. On appellera « Base des prototypes (B_p)» la base contenant les prototypes extraits de la base d'apprentissage, sa taille sera de $m * 1$ prototype.

5.1.2. Validation croisée – méthode du « leave-one-out »

On adopte comme stratégie de validation, la validation croisée et plus particulièrement la méthode du leave-one out, en fixant $\mu=1$. Pour cela, on crée k sous-bases extraites de la base de données, chaque sous-base, B_i , étant composée d'un exemple des m défauts, elle contient donc m exemples au total. Chaque sous-base est unique. Aucune liste d'alarmes n'est commune à plusieurs sous-bases. On a les relations suivantes entre les bases :

- $B_i \cap B_j = \emptyset \quad \forall i \neq j$
- $B_i \cup B_a = B$

On construit à partir de ces sous-bases, la base d'apprentissage en réunissant $k-1$ sous bases. La base de validation est constituée de la sous-base non utilisée dans la base d'apprentissage :

$$B_a = B \setminus B_i$$

$$B_v = B_i$$

Les paramètres de l'algorithme sont calculés sur la base d'apprentissage (dans le cas de la méthode vectorielle par exemple, il s'agit des prototypes et des poids). Les performances de l'algorithme sont évaluées sur les exemples de la base de validation.

La procédure complète -création de la base d'apprentissage, création de la base de test, paramétrages de l'algorithme, classification des exemples de la base de validation - est

réalisée k fois, de manière à ce que chaque sous-base soit utilisée une fois comme base de validation.

Les performances sont ainsi calculées sur l'ensemble de la base de données, en vérifiant la contrainte que les données utilisées pour l'apprentissage ne servent pas à la validation.

5.2. Mesures de performances

5.2.1. Courbe COR

Nous allons utiliser des courbes COR (Caractéristique Opérationnelle du Résultat) afin d'évaluer la capacité d'une distance, ou d'une mesure de similarité à discriminer deux exemples provenant d'un même défaut, de deux exemples provenant de deux défauts différents.

La courbe COR mesure la performance d'un détecteur, c'est à dire d'un classifieur donnant une réponse du type « Vrai ou Faux ».

Nous disposons d'une mesure de distance entre deux éléments (prototypes, vecteurs d'alarmes, séquences d'alarmes ...) notée D . Cette distance peut être utilisée à des fins de détection en fixant un seuil T_d . Si la distance entre deux éléments est inférieure au seuil de détection T_d , les deux éléments sont détectés comme générés par le même défaut, sinon ils sont détectés comme générés par deux défauts différents.

A partir d'une base de données composée de $n = k * m$ éléments, il est possible de construire $n * (n - 1)/2$ couples d'éléments. $n * (k - 1)/2$ sont des couples formés de deux éléments générés par un même défaut et $m * (k - 1)/2$ sont des couples formés d'éléments générés par deux défauts différents. Si l'on compare des exemples et des prototypes de défauts, il est possible de former $n * m$ couples, dont n proviennent d'un même défaut et $n * (m - 1)$ proviennent de défauts différents.

Pour une distance D et un seuil de détection T_d , il est possible de calculer le pourcentage de couples formés d'éléments provenant d'un même défaut correctement détectés comme provenant d'un même défaut. Il s'agit du pourcentage de bonnes détections. Il est aussi possible de calculer le pourcentage de couples formés d'éléments provenant de défauts différents détectés comme provenant d'un même défaut. Il s'agit du pourcentage de fausses alarmes.

La courbe COR se construit en calculant pour plusieurs valeurs de T_d le pourcentage de fausses alarmes et de bonnes détections et en traçant le pourcentage de bonnes détections en fonction du pourcentage de fausses alarmes.

Dans le cadre de mesures de distance entre éléments et prototypes de défauts, nous prendrons soin de toujours coupler un vecteur de la base de validation avec un prototype calculé à partir de la base d'apprentissage.

La Figure 5-2 montre un exemple de courbe COR

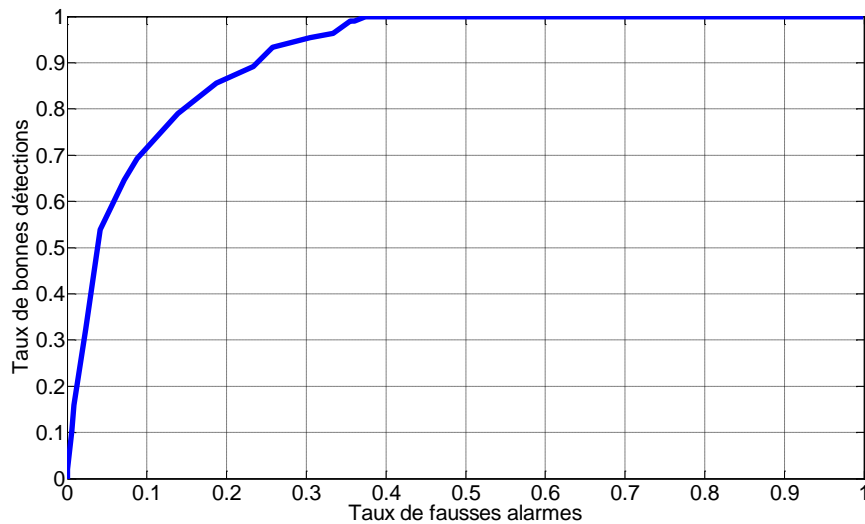


Figure 5-2 Exemple courbe COR

La courbe COR d'un détecteur idéal est la courbe qui passe par le point (0,1) (*Figure 5-3*)

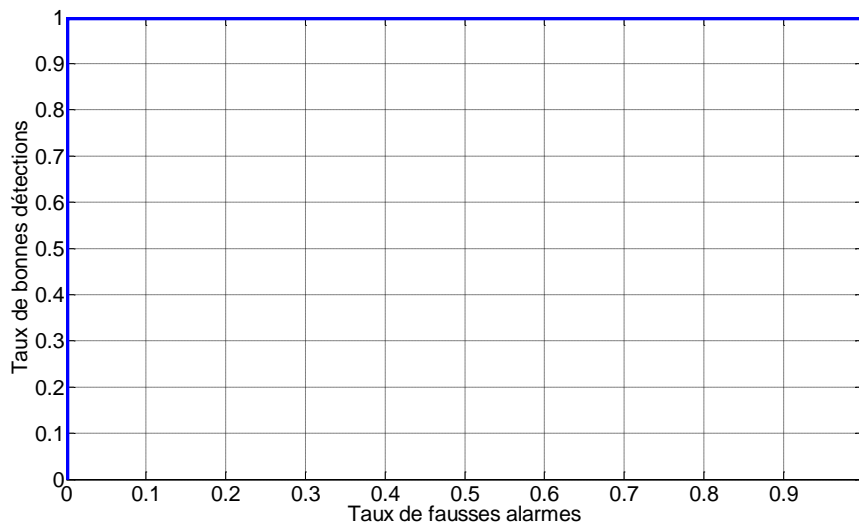


Figure 5-3 Exemple courbe COR détecteur idéal

La courbe COR d'un détecteur aléatoire (dont les performances sont équivalentes à tirer à pile ou face) est la droite d'équation $y=x$.

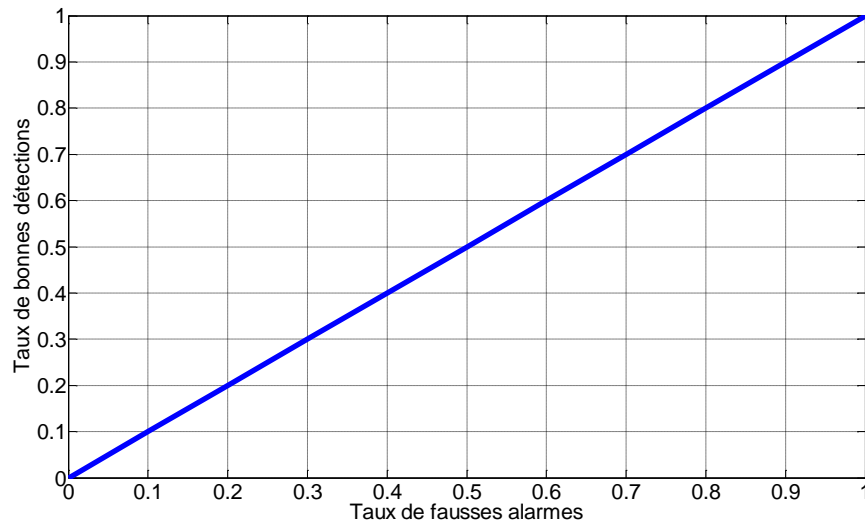


Figure 5-4 Exemple courbe COR détecteur aléatoire

La courbe COR permet de comparer les performances de plusieurs détecteurs. Dans le cas de coûts binaires (le coût d'une fausse alarme est égal au coût d'une non détection), le meilleur détecteur est celui dont la distance entre la courbe COR et le point (0,1) est la plus courte.

5.2.2. Matrices de confusion

Les résultats d'un classifieur peuvent être analysés à partir d'une matrice de confusion. Une matrice de confusion est une matrice de taille $m * m$ (m étant le nombre de classes à reconnaître, dans notre cas, les m défauts). Chaque case située sur la ligne numéro i et la colonne numéro j mesure le nombre d'exemples de classe i attribués à la classe j par le classifieur. Les exemples bien classés se trouvent donc sur la diagonale et les erreurs sont sur les termes en dehors de la diagonale. La matrice de confusion permet d'analyser en détail les erreurs faites par le classifieur.

Dans le cas où un classifieur peut fournir plusieurs réponses, ce qui est notre cas quand nous utilisons la stratégie de décision par seuillage d'une distance, nous proposons de modifier la matrice de confusion de la manière suivante.

Supposons que le classifieur attribue la liste $\{i, j, k\}$ comme classes possibles à un exemple de classe i . En ce cas, le chiffre 1 est ajouté dans les cases i, j et k de la matrice de confusion. La matrice de confusion étendue est analysée quand l'ensemble des exemples de la base de données a été classés. La somme des éléments de la ligne n° i divisée par le nombre d'exemples de classe i donne le nombre de diagnostics moyen pour la classe i . L'ensemble des cases de la ligne i dont la valeur est non nulle précise la liste exhaustive des classes confondues avec la classe i . La trace de la matrice divisée par le nombre d'exemples donne le pourcentage d'exemples dont la classe a été effectivement proposée dans la liste des classes possibles.

5.3. Conclusion

Nous avons présenté dans ce chapitre les outils de mesures de performances qui vont nous aider à évaluer l'efficacité des différents algorithmes de diagnostic présentés dans ce manuscrit. Une présentation des manipulations sur les bases est présentée en début de chapitre, avec notamment la présentation de la méthode de validation croisée qui permet de maximiser le nombre de tests. Puis deux outils majeurs ont été présentés, tout d'abord la courbe COR qui caractérise les performances d'un classifieur binaire. Nous avons ensuite présenté la matrice de confusion classiquement utilisée comme représentation graphique des décisions par un classifieur fournissant une classe unique, que nous avons étendu au cas des classifieurs pouvant proposer plusieurs classes possibles.

6. Analyse des performances des méthodes vectorielles

Dans ce chapitre, nous analysons les résultats obtenus avec la distance D_{vp} présentée au chapitre 4, en utilisant les différents outils de mesure de performances définis dans le chapitre 5. L'analyse se fait dans un premier temps en utilisant une base de données artificielle, puis les données extraites du simulateur du système gaz du CERN, présentées au chapitre 3.

6.1. Distances binaires

Afin d'évaluer les performances de notre méthode, nous la comparerons avec deux autres distances de la littérature.

6.1.1. La distance de Hamming

La première distance proposée est la distance de Hamming normée,

$$D_h(V, P^i) = \frac{\sum_{j=1}^N |v_j - p_j^i|}{N} \quad 6-1$$

Avec V et P^i , deux vecteurs binaires de longueur N , V étant le vecteur représentant la liste d'alarmes du défaut inconnu et P^i le vecteur représentant le défaut connu (il peut s'agir soit d'un exemple de défaut soit d'un prototype).

Chaque élément j est pondéré par $1/N$ et compte de manière identique dans le calcul de la distance. Les performances obtenues par cette distance très simple vont permettre d'évaluer la complexité du problème à résoudre, et plus particulièrement évaluer dans quelle mesure les exemples de chaque défaut varient de l'un à l'autre.

6.1.2. La distance définie dans (Charkaoui, et al., 2005).

Elle permet de calculer la distance entre deux vecteurs binaires. Elle utilise des prototypes calculés à partir d'un ensemble d'apprentissage, comme ceux définis dans le chapitre 5. La différence entre cette méthode et celle que nous proposons réside dans la construction des poids. (Charkaoui, et al., 2005) définit un vecteur dit de « déviation » qui mesure le degré de représentativité du défaut par le Prototype P^i . Les éléments du vecteur de déviation sont calculés de la manière qui suit :

$$\left\{ \begin{array}{l} \varepsilon_j^i = \frac{\sum_{q=1}^k v_j^{i,q}}{k} \text{ if } p_j^i = 0 \\ \varepsilon_j^i = \frac{k - \sum_{q=1}^k v_j^{i,q}}{k} \text{ if } p_j^i = 1 \end{array} \right. \quad 6-2$$

Avec p_j^i et $v_j^{i,q}$ tels que défini au paragraphe 4.2. Le terme ε_j^i est égal au terme $1-\alpha_j^i$, que nous avons défini. Il exprime le degré de représentativité de chaque alarme j dans le prototype associé au défaut i . On définit la distance D_{ch} entre le vecteur de défaut noté V et le prototype P^i à l'aide du vecteur de déviation de la façon qui suit :

$$D_{ch}(V, P^i) = R(V, P^i) - R(P^i, P^i) \quad 6-3$$

Avec

$$R(V, P^i) = \sum_{j=1}^N \left[\left(|v_j - p_j^i| \right) \bar{\varepsilon}_j^i + \left(1 - |v_j - p_j^i| \right) \varepsilon_j^i \right] \quad 6-4$$

Et

$$R(P^i, P^i) = \sum_{j=1}^N \varepsilon_j^i \quad \text{and} \quad \bar{\varepsilon}_j^i = 1 - \varepsilon_j^i \quad 6-5$$

Cette distance est équivalente à la distance D_{vp} non normée, en maintenant le terme β_i^j à la valeur 0.

6.2. Analyse des performances en utilisant une base de données artificielle

6.2.1. Création d'une base artificielle

Afin d'évaluer les performances des différentes distances, on construit, dans un premier temps, une base de données artificielle. Cette base est construite de façon à être la plus représentative du processus de génération d'alarmes du système de contrôle du CERN. Elle est composée de $m=13$ défauts et $k=12$ exemples par défaut. Les vecteurs d'alarmes sont composés de $N=77$ alarmes générées de la manière suivante :

- 50 alarmes aléatoires avec une probabilité d'apparition de 0.5.
- 13 alarmes aléatoires avec une probabilité d'apparition égale à 0.8.
- 13 alarmes déterministes et spécifiques au défaut, c'est-à-dire se déclenchant systématiquement (probabilité d'apparition égale à 1) et uniquement dans le défaut associé.

Cette base d'alarmes représente des cas réalistes et complexes, auxquels peuvent être confrontés les opérateurs de supervision. 2/3 des alarmes sont complètement aléatoires et n'apportent pas d'information pour le diagnostic. 1/6 se déclenchent fréquemment (80% du temps), mais sont communes à tous les défauts. Seule 1/6 des alarmes sont pertinentes pour le diagnostic. Ainsi, pour une liste d'alarmes déclenchées contenant en moyenne 50×0.5 (alarmes purement aléatoires) + 13×0.8 (alarmes présentes dans la majorité des défauts) + 1 (l'alarme spécifique au défaut), soit 37 alarmes, seule la présence d'une alarme va permettre à l'opérateur de diagnostiquer le défaut. Le rapport signal utile sur bruit est donc très faible.

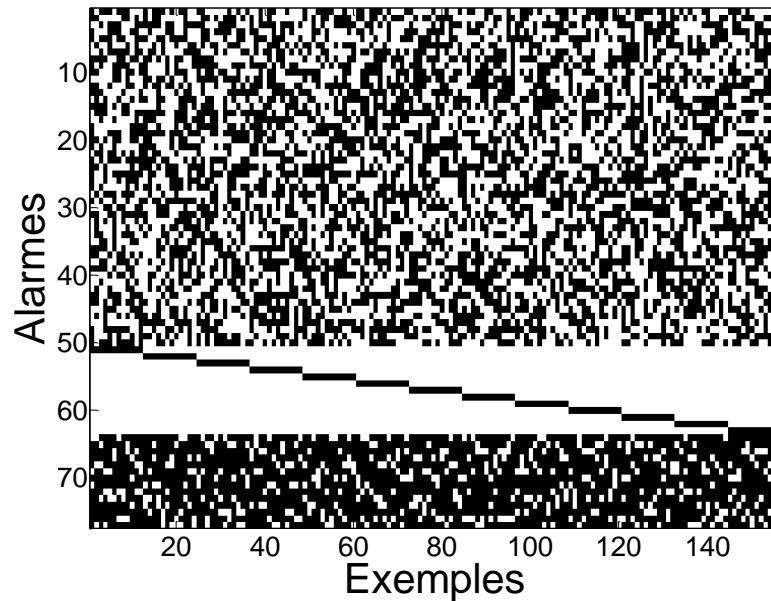


Figure 6-1 Base aléatoire

La Figure 6-1 montre l'exemple d'un jeu de données simulées. Les vecteurs de tous les défauts sont juxtaposés afin de créer une matrice où chaque colonne représente un vecteur V et chaque ligne une alarme. Les vecteurs sont groupés par défaut. Lorsque l'élément de la matrice est noir, cela correspond à une alarme mis à 1. Lorsqu'il est blanc, l'alarme est à 0.

Afin d'évaluer les performances des distances de manière réaliste, il est nécessaire que chaque jeu de simulation soit différent. Nous allons analyser la variabilité de la base de données générée et vérifier que nous avons réussi à produire artificiellement une base de données composée d'exemples différents pour un même défaut. Pour ce faire, nous calculons de manière empirique la fréquence d'apparition de chaque alarme au sein d'un même défaut, sur un jeu de simulation. Les résultats sont présentés sous forme de boîtes à moustaches. Dans chaque boîte, le trait central représente la médiane, les extrémités de chaque boîte sont les 25ème et 75ème quartile, les extrémités des segments représentent les valeurs extrêmes, les points aberrants sont représentés par des croix.

La courbe de gauche présente la fréquence d'apparition des 50 premières alarmes, dont la probabilité théorique d'apparition a été réglée à 0.5. La courbe de droite représente la fréquence d'apparition des 13 dernières alarmes, dont la probabilité d'apparition a été réglée à 0.8. La boîte de gauche est construite à partir de 13x50 points, celle de droite avec 13x13 points.

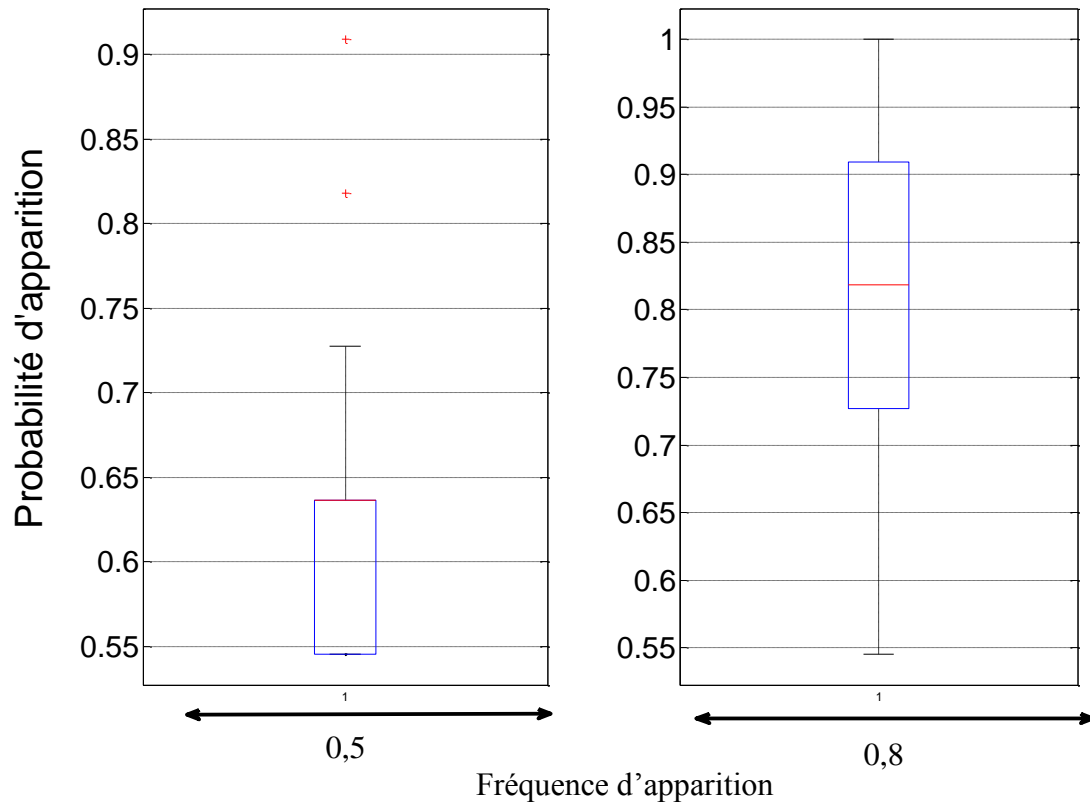


Figure 6-2 Boîtes à moustaches de la fréquence théorique d'apparition d'alarmes

Du fait du faible nombre d'exemples par défaut (11 sur 12 sont utilisés pour calculer la fréquence d'apparition, ce qui correspond à l'estimation du α_j^i quand la méthode de validation du « leave one out » est utilisée), la fréquence d'apparition estimée des 50 premières alarmes se distribue entre des valeurs comprises entre 0.55 (11/12) et 0.73. De même, la fréquence d'apparition des 13 dernières alarmes se distribue entre 0.55 et 1. Cela montre que la base de données générée a une variabilité représentative d'une situation réelle. Chaque alarme n'a pas une fréquence d'apparition fixe de 0.5 ou de 0.8. Celle-ci varie pour chaque alarme, pour chaque défaut et pour chaque jeu de simulation, et ceci de manière conséquente.

Afin de tester les capacités en rejet de la méthode, 12 exemples d'un nouveau défaut (défaut numéro 14) sont générés. Les alarmes aléatoires sont construites de la même façon que pour les 13 autres défauts. Seule la valeur des 13 alarmes déterministes change. Un exemple de défaut inconnu est représenté par d alarmes déterministes de valeur égale à 1. Afin de faire varier la similarité du défaut 14 avec les 13 autres défauts, la valeur de d peut varier entre $d=2$ (très similaire. Une seule alarme différencie le défaut 14 du défaut 1) à $d=13$ (très différent). Un jeu de simulation de la nouvelle base avec $d=3$ est présenté Figure 6-3.

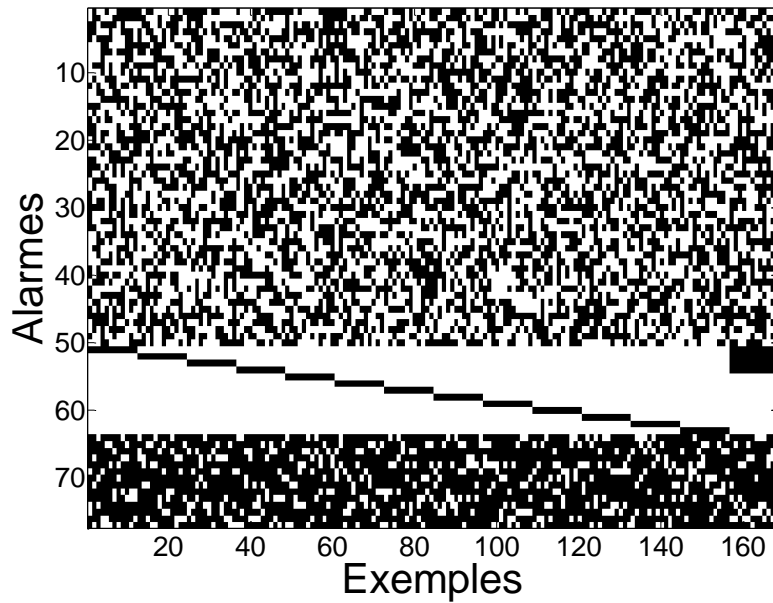


Figure 6-3 Base aléatoire avec défaut à rejeter

6.2.2. Résultats obtenus

6.2.2.1. Analyse des poids sur un jeu de données

Comme nous l'avons présenté dans le chapitre 4, les poids calculés à partir de la base d'apprentissage peuvent être utilisés comme informations sur le système et présentés à l'opérateur sous forme de trois matrices : la matrice des poids w_i^j , la matrice des poids $(2 * \alpha_j^i - 1)$, la matrice des poids $(1 - \beta_j^i)$.

Les figures suivantes présentent les matrices de poids extraites d'un jeu de données simulées. Les défauts sont rangés en colonnes de 1 à 13. La gradation de couleurs varie de bleu pour les valeurs proches de 0 à rouge pour les valeurs proches de 1.

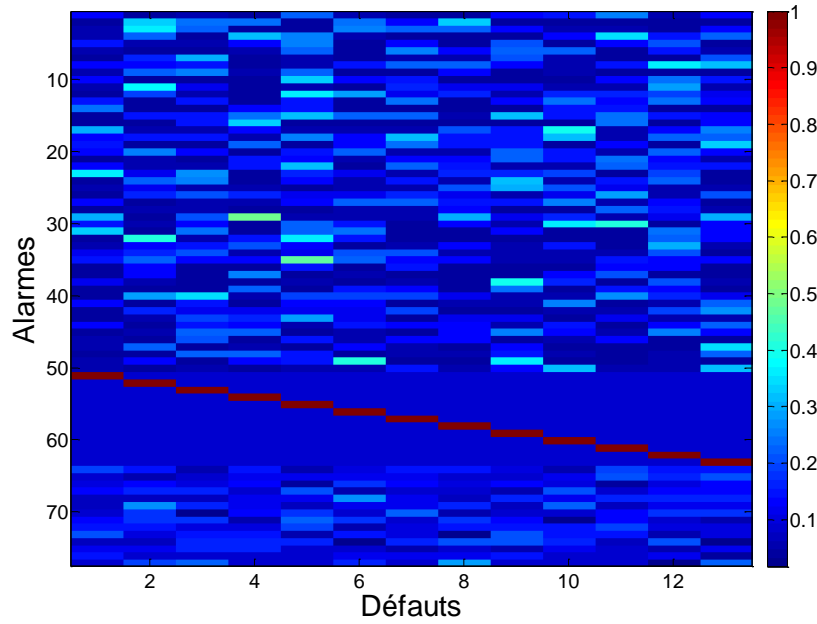


Figure 6-4 Poids $w_j^i = (2 * \alpha_j^i - 1) * (1 - \beta_j^i)$ des défauts de la base artificielle

La première image représente les poids w_i^j . Elle met en évidence les alarmes permettant de diagnostiquer un défaut parmi l'ensemble des défauts de la base. L'analyse combinée des deux dernières images présentant $(2 * \alpha_j^i - 1)$ et $(1 - \beta_j^i)$ permet de différencier les alarmes non discriminantes pour les défauts car purement aléatoires des alarmes dont la valeur est globalement identique pour tous les défauts et qui sont donc du bruit pour le diagnostic.

Sur la matrice des poids w_i^j , les alarmes de 1 à 50 et de 65 à 77 apparaissent avec des valeurs de poids faibles, signifiant à l'expert qu'elles ne sont pas pertinentes pour élaborer le diagnostic. La matrice des poids montre clairement que les alarmes de 51 à 64 sont les alarmes qui rendent le système diagnosticable. Chaque défaut de 1 à 13 est spécifié par une unique alarme sur lequel l'opérateur doit se focaliser pour diagnostiquer le défaut.

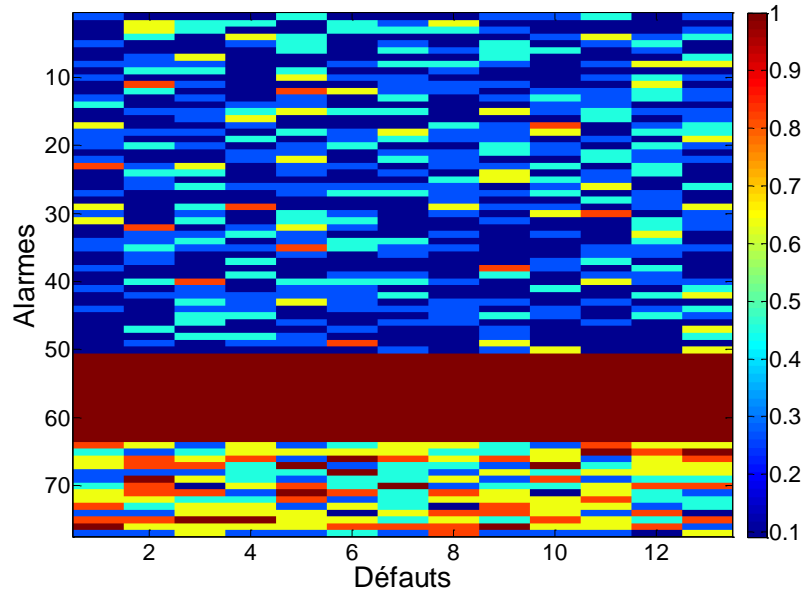


Figure 6-5 Poids $(2 * \alpha_j^i - 1)$ sur la base artificielle

La matrice $(2 * \alpha_j^i - 1)$ met en évidence les alarmes déterministes de la base. Celles-ci apparaissent en rouge alors que les alarmes de 1 à 50 apparaissent dans des couleurs bleues. Notons que la variabilité du jeu de données, avec des valeurs de α_j^i fortement variantes d'une alarme et d'un défaut à l'autre, sur les alarmes aléatoires, s'observe par la variation de couleurs générée. Certaines alarmes entre 1 et 50 apparaissent en rouge, ce qui signifie que, pour ce jeu de données, ces alarmes ont pris une valeur binaire constante. Comme nous l'avons déjà dit, cette variabilité observée sur les données permet de tester les méthodes sur des exemples différents d'un jeu de données à l'autre, ce qui permet une analyse des performances plus fiable.

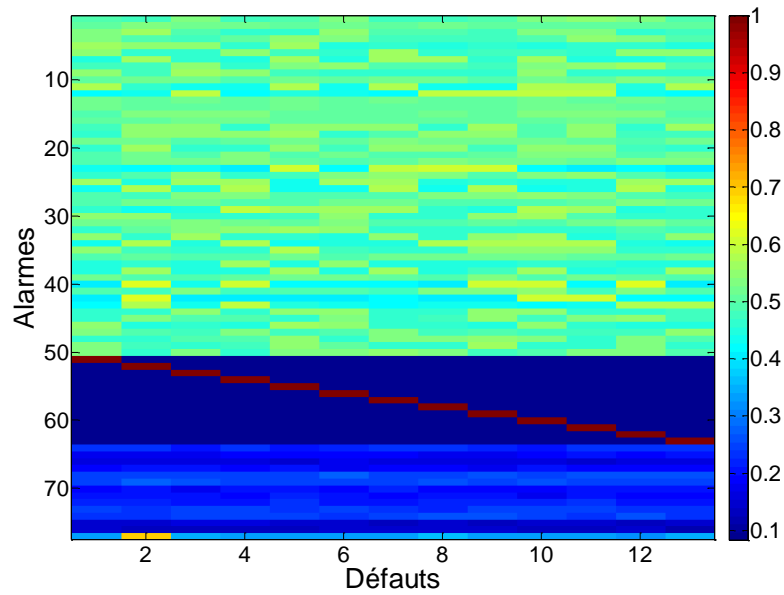


Figure 6-6 Poids $(1 - \beta_j^i)$ base artificielle

La matrice $(1 - \beta_j^i)$ met en évidence les alarmes dont la valeur est partagée par un grand nombre de défauts. Les alarmes de 65 à 77 apparaissent ainsi dans des couleurs bleutées. Il s'agit justement de celles qui sont partagées par l'ensemble des défauts avec une fréquence relativement élevée. Elle met également en relief les alarmes 51 à 64, dont la valeur prise au sein d'un défaut est systématiquement l'inverse de celle des autres défauts.

6.2.2.2. Performances en diagnostic

6.2.2.2.1. Localisation d'un défaut contenu dans la base d'apprentissage

i. Courbe COR

On analyse dans un premier temps la capacité des différentes distances à discriminer les couples de vecteurs et de prototypes provenant d'un même défaut de ceux provenant de défauts différents. Lorsque l'on réalise un diagnostic à l'aide de la distance de Hamming (D_h), deux possibilités s'offrent à nous. La première consiste à mesurer la distance entre un couple de deux vecteurs (vecteur, vecteur), la deuxième à mesurer la distance entre des couples formés d'un vecteur et d'un prototype (vecteur, prototype). Dans ce qui suit, on effectuera l'étude en calculant les résultats des deux manières afin d'évaluer l'apport de la synthèse des exemples en un prototype. Pour les distances D_{ch} et D_{vp} , de par leur définition, les distances seront mesurées sur des couples de types (vecteur, prototype).

Lorsque les distances sont utilisées sur des couples (vecteur, prototype), la différence de performances entre les distances D_h , D_{ch} et D_{vp} ne provient que des poids appliqués aux alarmes, les prototypes étant calculés de manière identique. Rappelons enfin que, grâce à la

méthode de validation du leave-one-out utilisée, le prototype, ainsi que les poids, sont toujours calculés en excluant le vecteur auquel il est comparé.

La figure ci-dessous représente les courbes COR des quatre distances, avec, en rouge la courbe COR de la distance D_h entre couple (vecteur, vecteur), en bleue celle de la distance D_h entre couple (vecteur, prototype), en noir la courbe de COR de D_{ch} et enfin, en vert, la courbe de D_{vp} .

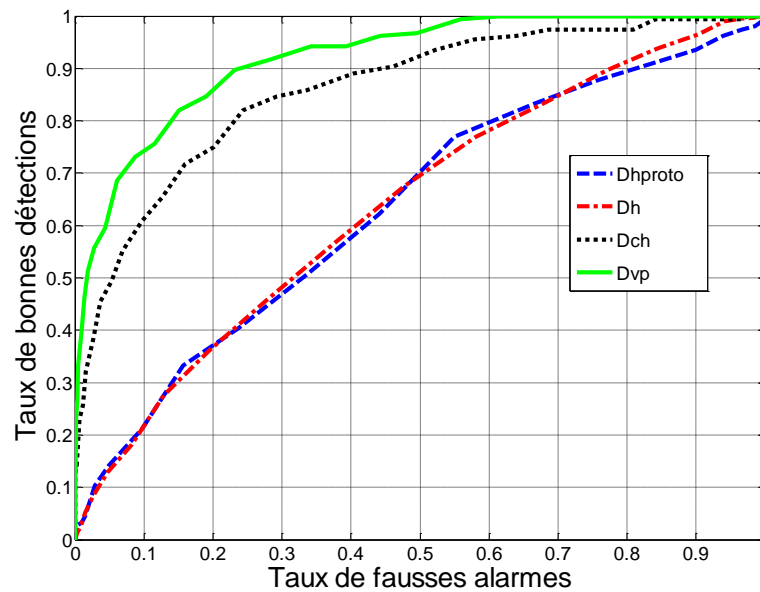


Figure 6-7 Courbe COR sur base artificielle

La distance de Hamming donne, comme attendu, les moins bons résultats avec des performances proches de l'aléatoire. Les performances de la distance D_h sont équivalentes, que la distance soit mesurée entre deux vecteurs ou entre un vecteur et un prototype. Cela montre que la synthèse des vecteurs en un prototype ne suffit pas à améliorer les performances en diagnostic. Bien qu'ils permettent de synthétiser plusieurs défauts en un unique vecteur en éliminant les alarmes considérées comme du bruit, il ne constitue qu'un premier pas dans la description d'un défaut. Une distance purement binaire n'est pas suffisante dans ces situations car la distance est calculée à partir des 77 alarmes, à parts égales. L'information utile, portée par les alarmes de 51 à 64, se retrouve donc noyée dans le bruit généré par les autres alarmes, non discriminantes. Ceci montre la nécessité de développer des distances plus élaborées afin de répondre à cette problématique. Ces mauvais résultats montrent également la variabilité de la base artificielle et la difficulté de réaliser un diagnostic à partir de cette base.

Les courbes COR de la distance D_{ch} et de la distance D_{vp} permettent d'obtenir de bien meilleurs taux de bonnes détections pour des taux de fausses alarmes équivalents. L'utilisation des poids associés aux alarmes permet de donner plus d'importance aux

informations utiles pour le diagnostic, et donc d'augmenter implicitement le rapport signal sur bruit de la distance. Dans le cas de D_{ch} , l'information utilisée est portée par les alarmes 51 à 64 et 65 à 77. L'influence des alarmes 1 à 50 est diminuée grâce au vecteur de déviation. La distance D_{vp} obtient les meilleures performances car l'information utilisée par la distance se concentre, via les poids, sur les alarmes 51 à 64.

ii. Classification par la méthode du plus proche voisin

L'analyse des performances est maintenant réalisée à partir de la méthode du plus proche voisin. Chaque exemple de la base de validation est affecté à la classe de son plus proche voisin dans la base d'apprentissage. Le pourcentage d'exemples bien classés sert de critère de performance. Les trois distances D_h , D_{ch} et D_{vp} à partir des poids calculés en « seul contre tous » sont évaluées ainsi que la stratégie des méthodes en deux étapes proposées au chapitre 4. On rappelle le principe de cette méthode : les deux plus proches voisins au sens de la distance D_{vp} avec les poids calculés suivant la stratégie du « seul contre tous » sont sélectionnés, puis le plus proche voisin au sens de la distance D_{vp} avec les poids recalculés suivant la stratégie du « un contre l'autre » est finalement sélectionné.

Afin d'évaluer les performances sur un grand nombre de tests, 20 jeux de simulation sont générés et le taux de réussite est calculé sur chaque jeu. Les résultats sont présentés à l'aide de boîtes à moustaches.

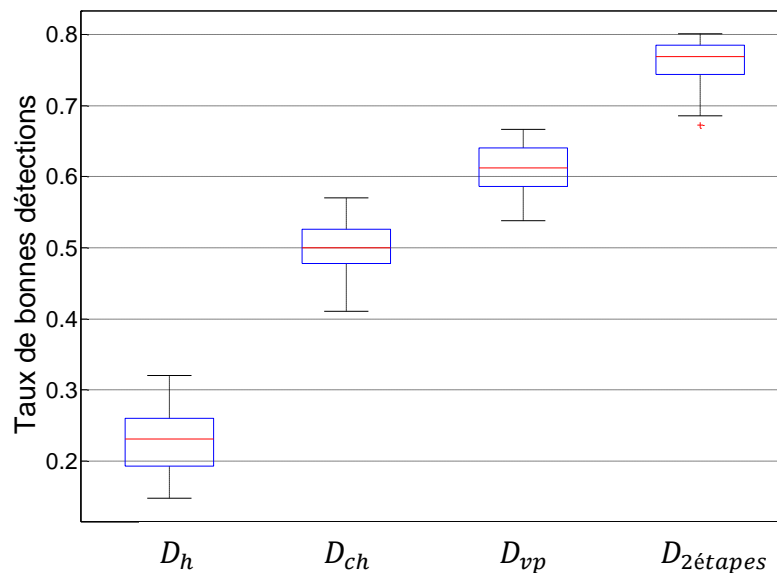


Figure 6-8 Boîtes à moustaches des résultats des différentes distances

Les distances obtiennent le même classement en performance que celui donné par la courbe COR. La distance de Hamming est la distance donnant les résultats les plus faibles avec un pourcentage de biens classés de 24%, la distance D_{ch} donne quant à elle de meilleurs résultats que D_h , grâce à l'ajout des pseudo-poids du vecteur de déviation. Ils vont limiter

l'influence des alarmes purement aléatoires. D_{vp} donne de meilleurs résultats en moyenne que D_{ch} avec un taux de réussite de l'ordre de 61% soit environ trois fois mieux que D_h , et une amélioration de l'ordre de 22% comparée à la distance D_{ch} . Ceci montre l'intérêt de prendre en compte les exemples des autres défauts dans le calcul des poids. Enfin, la distance D_{vp} en deux étapes avec un calcul des poids « seul contre tous » puis « un contre l'autre » est la méthode donnant les meilleurs résultats, avec un taux de réussite moyen de 77%. La plupart des défauts étant très proches avec une seule alarme spécifique, le calcul des poids lors de la deuxième étape renforce l'influence des deux alarmes spécifiques. Notons que pour chaque jeu de simulation, l'ordre de classement obtenu a toujours été le même : D_h , D_{ch} , D_{vp} , $D_{2\text{étapes}}$.

Afin d'analyser plus en détail l'efficacité de la méthode en deux étapes, nous comparons le nombre d'exemples bien classés par la méthode en deux étapes avec le nombre de fois où le défaut à reconnaître était présent parmi les deux défauts retenus lors de la première étape. On évalue en fait la capacité du classifieur à reconnaître le bon défaut parmi deux défauts possibles.

L'analyse est effectuée sur les 20 jeux de simulation. Les résultats sont présentés Figure 6-9, avec, en abscisse, le nombre de fois où le défaut à reconnaître était présent dans la liste de deux défauts et en ordonnée, le nombre d'exemples bien classés par la méthode en deux étapes. Le nombre d'exemples à classer par jeu est de 156. La figure contient les 20 points des 20 jeux de simulation. La droite $y=x$ permet une meilleure visualisation des résultats. Tout point situé sur la droite correspond à un jeu de test où tous les défauts dont le diagnostic était présent dans la liste des deux défauts ont été correctement diagnostiqués lors de la 2^{ème} étape. Les points sous la courbe correspondent aux jeux de test où des erreurs de diagnostic ont été faites lors de la 2^{ème} étape.

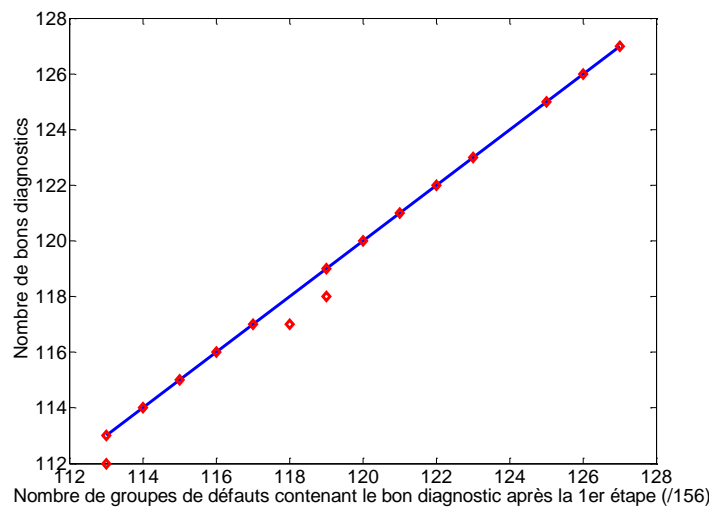


Figure 6-9 Nombre de groupes de défauts contenant le bon diagnostic après 1^{er} étape sur base artificielle

L'analyse du graphique montre que, dans la quasi-totalité des cas, si le bon diagnostic est effectivement contenu dans les deux propositions faites par D_{vp} lors de la première étape, alors la réponse du classifieur en deux étapes sera correcte. Le taux de reconnaissance de la 2^{ème} étape est de 99.7%. Cela démontre la forte capacité de différenciation de la méthode. On peut également en déduire que le facteur limitant est le premier filtrage réalisé lors de la première étape où les poids sont calculés à partir de toute la base d'apprentissage (« un contre tous »).

Afin de mieux comprendre la méthode en deux étapes, nous étudions les poids d'un défaut lorsqu'ils sont calculés dans un premier temps avec toute la base en utilisant la stratégie du « seul contre tous » puis, dans un second temps, lorsque la base est réduite à deux défauts, avec la stratégie du « un contre un ». Le défaut considéré est le défaut 13. Les poids présentés sont les poids normés : ils sont divisés par la somme des poids du prototype et représentent donc le facteur de pondération avec lequel les alarmes interviennent dans le calcul de la distance. Les poids sont représentés dans la figure ci-dessous sous forme d'image. La partie de gauche correspond aux poids relatifs calculés en « seul contre tous », celle de droite en « un contre un ». Le défaut 13 est comparé au défaut 10, qui est le 2^{ème} candidat sélectionné lors de la 1^{ère} étape.

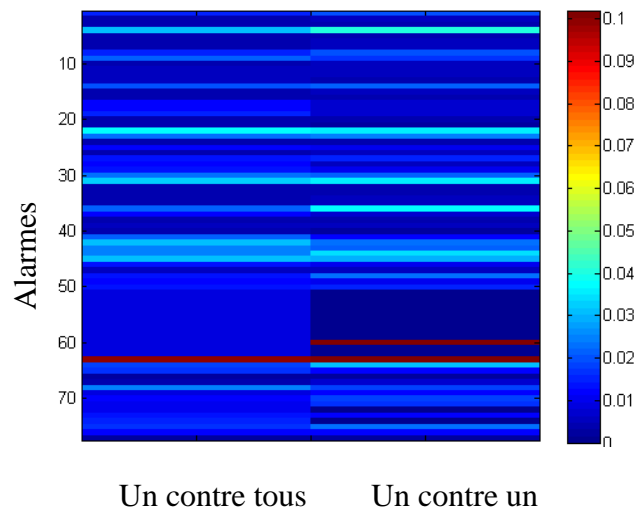


Figure 6-10 Poids du défaut 13 en seul contre tous et un contre un

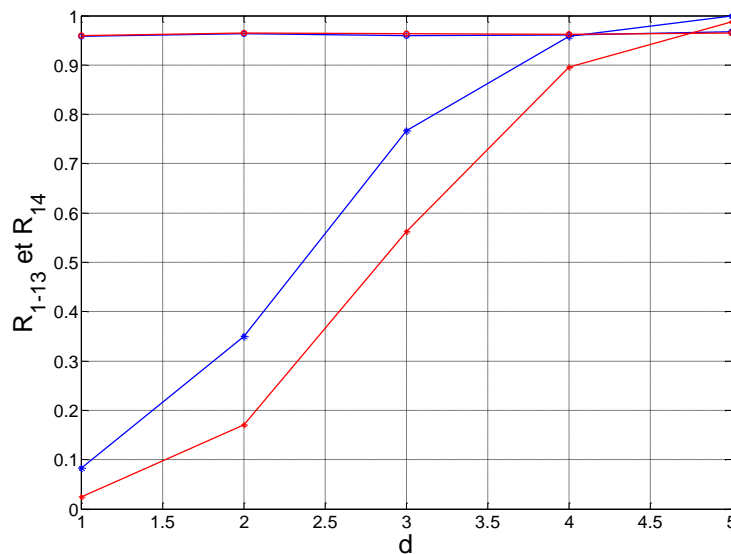
Lors de la première étape, l'alarme 63 est considérée comme la plus discriminante et intervient avec un poids de 0.1 dans le calcul de la distance. Son poids relatif est environ 8 fois supérieur à celui attribué par la distance de Hamming (0.013). Lors de la 2^{ème} étape, l'alarme 60, spécifique au défaut 10, voit son poids fortement augmenter et atteindre une valeur équivalente à l'alarme 63. Le calcul de la distance est réalisé principalement à partir de ces deux alarmes, qui comptent pour 20% dans le calcul de la distance. En contrepartie, d'autres alarmes, non pertinentes pour discriminer les 2 défauts, voient leur poids diminuer.

Lors du calcul des poids en « un contre l'autre », en plus de vérifier la présence de l'alarme 63 représentative du défaut 13, l'algorithme se concentre également sur l'absence nécessaire de l'alarme 60 pour que le défaut soit diagnostiqué comme étant le défaut 13.

6.2.2.2.2. Rejet de défauts non contenus dans la base d'apprentissage

Après avoir analysé les comportements de différentes distances lorsque le défaut que l'on cherche à diagnostiquer est effectivement contenu dans la base d'apprentissage, on va étudier maintenant le comportement des distances lorsque le prototype du défaut que l'on cherche à diagnostiquer n'est pas contenu dans la base. L'analyse est effectuée à partir de la base de données augmentée du 14^{ème} défaut. 20 jeux de simulation différents contenant 14 défauts sont réalisés. Le premier jeu est utilisé pour déterminer un seuil de rejet noté T_r . Ce seuil est fixé de telle sorte que sur la base contenant les défauts de 1 à 13 le pourcentage de défauts rejetés soit de 4% (on accepte donc ici 4% des défauts 1 à 13 rejetés par erreur). Une fois le seuil T_r défini, nous calculons sur chacune des 19 bases restantes le pourcentage d'exemples du défaut 14 correctement rejetés (noté R_{14}), ainsi que le pourcentage d'exemples de défauts 1 à 13 non rejetés (noté R_{1-13}). Les différents pourcentages sont calculés pour une valeur de d (voir paragraphe 6.2.1) variant entre 2 à 5. On trace les différents résultats sur un graphique représentant R_{1-13} et R_{14} en fonction du nombre d . Les résultats sont représentés à l'aide de deux courbes, la courbe bleue représente les résultats obtenus avec D_{ch} et la rouge ceux obtenus avec D_{vp} . R_{1-13} est représenté par le signe « o » et R_{14} par le signe « * ». Chaque point correspond à la médiane des valeurs sur les 20 jeux de simulations.

Rappelons que dans la procédure de rejet, les poids de D_{vp} sont calculés avec $\beta_j^i=0$.



R_{1-13} (*) et R_{14} (o), D_{vp} (bleu) and D_{ch} (rouge)

Figure 6-11 Pourcentage de bons rejets en fonction de d

La valeur de R_{14} (pourcentage d'exemples correctement rejetés) augmente avec d , le nombre d'alarmes déterministes qui différencient les défauts 1 à 13. Détecter un nouveau défaut lorsque $d = 2$ c'est-à-dire deux alarmes différencient les défauts, est quasiment impossible, (le pourcentage d'exemples correctement rejetés est de 10%). Le défaut 14 est alors trop similaire aux défauts contenus dans la base. Le taux maximum de 100% de bons rejets est atteint lorsque $d=5$. Notons également que le pourcentage d'exemples de défauts 1 à 13 non rejetés est lui constant et en accord avec le paramétrage réalisé avec le jeu utilisé pour définir T_r . On constate que les résultats obtenus avec D_{vp} sont meilleurs que ceux fournis par D_{ch} . En effet, pour tout d , le pourcentage d'exemples correctement rejetés de la distance D_{vp} est supérieur à celui de D_{ch} . Les poids en rejet de D_{vp} sont calculés sans la partie $(1 - \beta_j^i)$. Ainsi, l'information utilisée par les deux distances est similaire : la probabilité apparition d'une alarme. Seule la manière d'intégrer l'information dans la distance diffère. La prise en compte de cette information sous forme de pondération normée dans le cas de D_{vp} permet l'obtention de meilleures performances.

6.3. Analyse des performances en utilisant la base de données du CERN

Dans cette partie, nous étudions et comparons les résultats obtenus pour chaque distance sur la base de données fournis par le simulateur des processus gaz présenté dans le chapitre 3. Le nombre de défaut est ici de 13 et le nombre d'exemples disponibles est de 6 par défaut. Chaque exemple est obtenu lorsque le procédé est dans un état de fonctionnement différent. Nous avons répertorié 460 alarmes potentiellement déclenchables par le simulateur mais seules 97 alarmes différentes ont été déclenchées par le système sur l'occurrence des défauts considérés. Chaque défaut est donc représenté par un vecteur de taille $N=97$.

6.3.1. Analyse des alarmes et des poids

Afin d'étudier les vecteurs de défauts générés par le simulateur du système gaz, on calcule la fréquence d'apparition des alarmes pour chaque défaut. On représente les valeurs obtenues sous forme d'une image. Chaque colonne représente un défaut et chaque ligne une alarme.

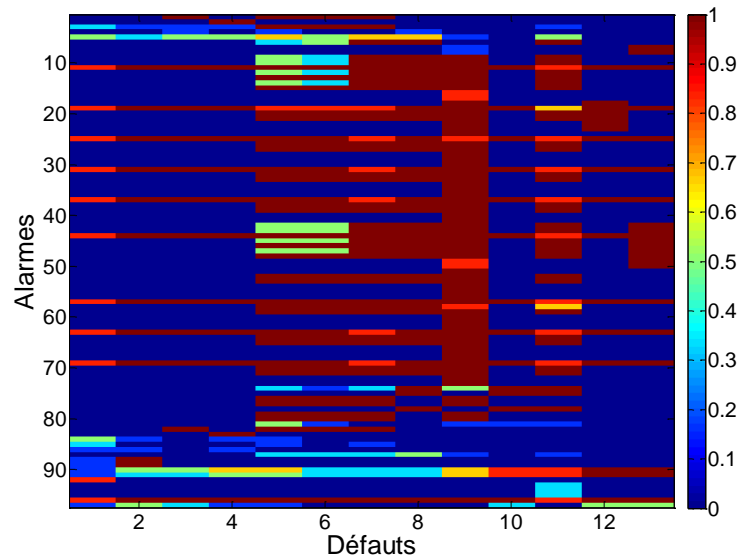


Figure 6-12 Fréquence d'apparition des alarmes en fonction du défaut

Les alarmes ne se déclenchant jamais pour un défaut apparaissent en bleu foncé. Celles se déclenchant systématiquement apparaissent en rouge foncé. Celles qui se déclenchent environ une fois sur deux apparaissent en vert clair. Ces dernières sont relativement peu nombreuses. Seuls les défauts 5 et 6 en contiennent un nombre conséquent. Les alarmes déterministes, se déclenchant systématiquement en présence du défaut, sont beaucoup plus nombreuses. Cependant, elles sont fréquemment partagées par plusieurs défauts.

Analysons maintenant les matrices de poids déduites des exemples de défaut du simulateur. La Figure 6-13 montre les valeurs des paramètres $(2 * \alpha_j^i - 1)$ pour les 13 défauts de la base. La matrice met clairement en évidence les alarmes déterministes et celles qui sont aléatoires. Elle confirme le fait que la majorité des alarmes se déclenchent (ou ne se déclenchent pas) de manière systématique pour les exemples d'un même défaut.

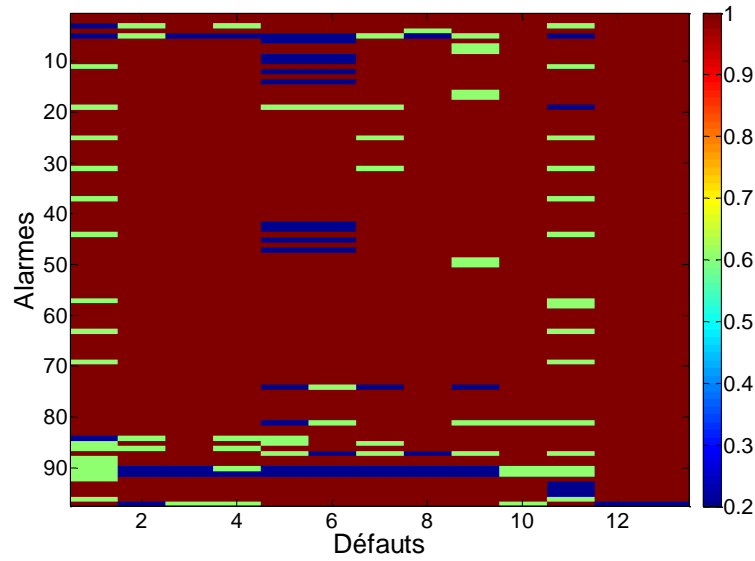


Figure 6-13 Poids $2\alpha_j^i - 1$

Étudions maintenant les valeurs de $(1 - \beta_j^i)$ calculées à l'aide de la méthode « seule contre tous ».

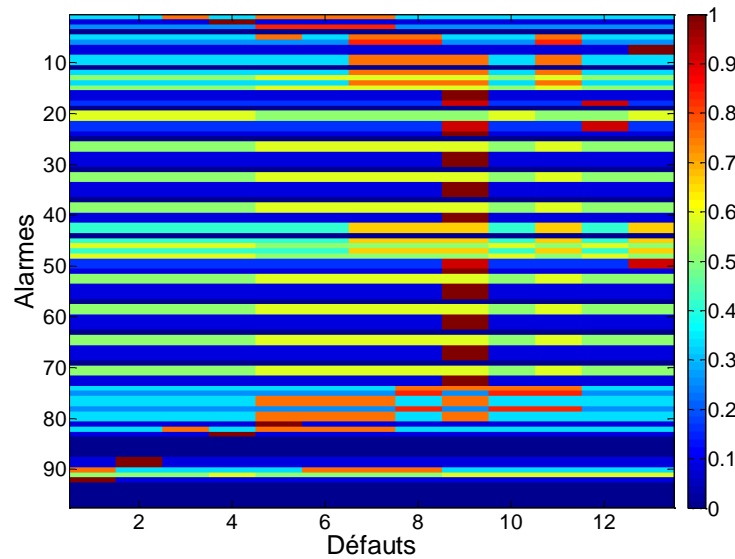


Figure 6-14 Poids des défauts $(1 - \beta_j^i)$

L'image met en évidence les alarmes qui prennent, sur l'occurrence du défaut, une valeur contraire à celle prise dans la majorité des autres défauts. On constate, qu'hormis le défaut 9, les défauts contiennent relativement peu ce type d'alarmes. Ceci est compréhensible car cela correspond à une alarme prenant la valeur 1 pour un défaut donné et la valeur 0 pour la majorité des autres défauts (ou inversement). Compte tenu du nombre de défauts (13), il est statistiquement compréhensible qu'une alarme prenne la même valeur dans plusieurs défauts.

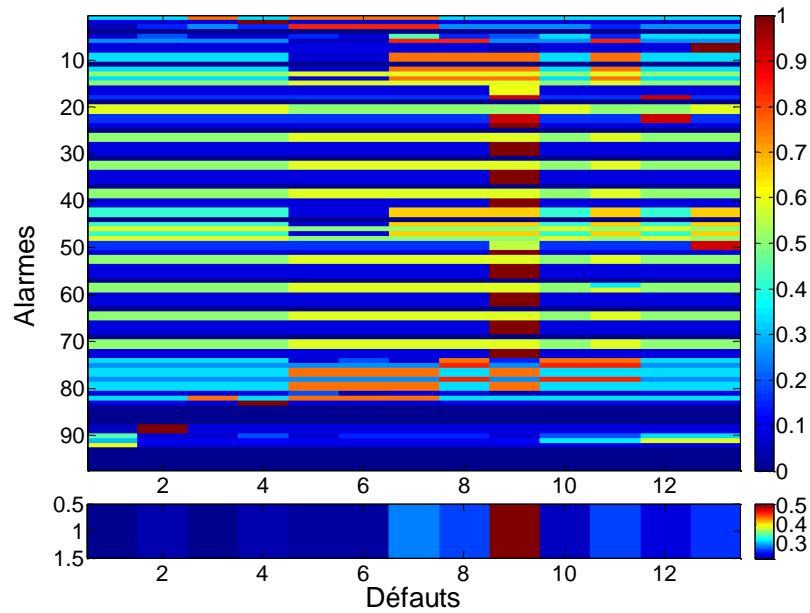


Figure 6-15 Poids des défauts calculés $(2 * \alpha_j^i - 1) * (1 - \beta_j^i)$

La Figure 6-15 montre les poids calculés à l'aide de α_j^i ainsi que β_j^i selon l'équation développée au chapitre 4: $(2 * \alpha_j^i - 1) * (1 - \beta_j^i)$. L'image centrale représente les poids pour chaque alarme et chaque défaut. L'image du dessous représente le poids moyen du défaut, moyenné sur toutes les alarmes. On constate que le défaut 9 sort du lot, avec un poids moyen beaucoup plus élevé, ce qui signifie qu'il possède un nombre important d'alarmes spécifiques, ce qui devrait le rendre très aisément diagnostiquable. L'analyse de l'image centrale met en évidence que la plupart des défauts contient des alarmes spécifiques, mais en très faible nombre. L'image est fortement similaire à celle obtenue avec $(1 - \beta_j^i)$ car le nombre d'alarmes aléatoires, pour lesquels α_j^i prend une valeur proche de 0, est faible.

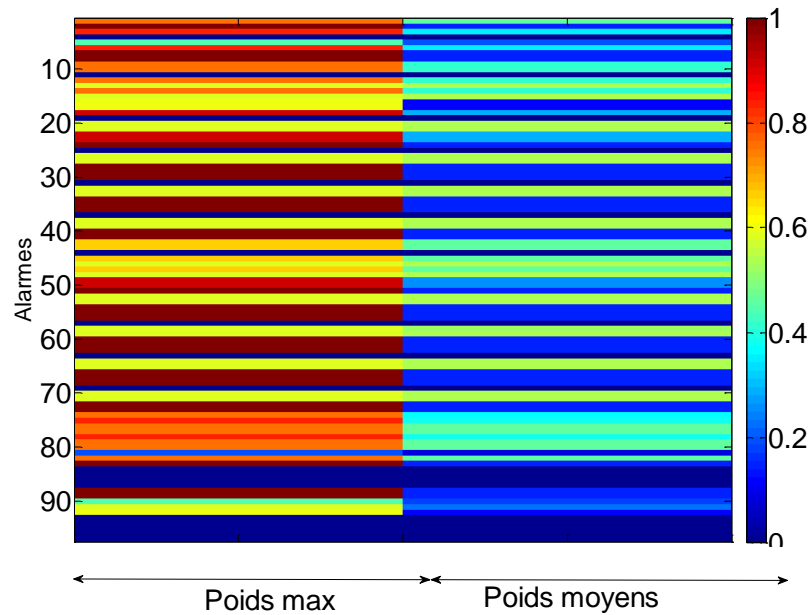


Figure 6-16 Poids maximum contre poids moyens

La matrice des poids est maintenant analysée ligne par ligne. La Figure 6-16 présente, sur la partie de gauche, le poids maximal prise par une alarme sur les 13 défauts, c'est à dire la valeur maximale des poids par ligne. L'image de droite représente le poids d'une alarme moyennée sur les 13 défauts (la moyenne des poids par ligne). Cette figure informe sur l'intérêt global de l'alarme pour le diagnostic. Ainsi, toute alarme de poids moyen et de valeur maximale faible correspond à une alarme non pertinente pour le diagnostic. Elle n'est spécifique à aucun des 13 défauts. On dénombre 17 de ces alarmes, ce qui signifie que 18% (17/97) des alarmes générées par le système de contrôle semblent inutiles pour diagnostiquer les 13 défauts. Les alarmes ayant un fort poids maximal mais un faible poids moyen sont les alarmes qui sont spécifiques à très peu de défauts. Elles sont très utiles pour le diagnostic. Ce sont celles-là qu'il faudra mettre en avant lors de la localisation du défaut. La moitié environ des alarmes sont des alarmes qui permettent de différencier les alarmes entre elles, cependant elles sont mal distribuées sur les 13 défauts avec une grande majorité déclenchées pour le défaut numéroté 9.

6.3.2. Performances en diagnostic

6.3.2.1. Localisation d'un défaut contenu dans la base d'apprentissage

6.3.2.1.1. Courbes COR

On analyse dans un premier temps les courbes COR des distances D_{vp} (vert), D_{ch} (noir) et D_h utilisées avec des prototypes (bleu) et sans prototype (rouge). Les poids de D_{vp} sont calculés en « seul contre tous ».

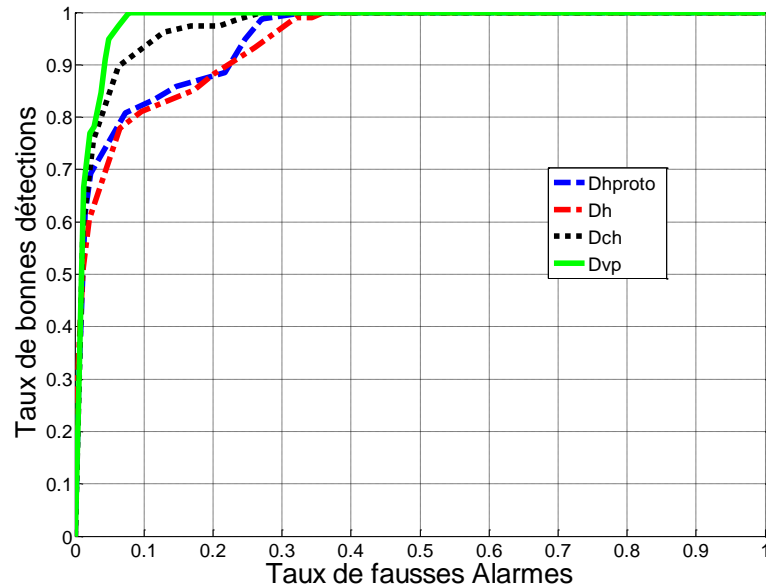


Figure 6-17 Courbe COR base CERN

L'analyse de la courbe COR, nous montre le même classement en performances que lors de l'analyse faite sur la base artificielle. En effet, les meilleurs résultats sont obtenus avec la distance D_{vp} , les moins bons résultats sont obtenus avec les distances D_h que le vecteur soit comparé à un autre vecteur ou à un prototype. Ceci montre que le réel apport de la distance réside dans le calcul des poids et leur utilisation lors de calcul de la distance. D_{ch} se situe entre les deux distances. Notons les bonnes performances obtenues par D_{vp} qui atteint un taux de bonnes détections de 95% pour un taux de fausses alarmes de 5%. Par comparaison, pour un taux de fausses alarmes de 5%, on obtient seulement 82% de bonnes détections avec la distance D_{ch} et enfin un taux de 70% pour la distance D_h . Ces résultats sont en accord avec l'étude des performances avec la base artificielle.

6.3.2.1.2. Classification par la méthode du plus proche voisin

Le Tableau 6-1 représente les résultats obtenus en utilisant la méthode du plus proche voisin en utilisant les trois distances puis la méthode en deux étapes pour les 78 exemples.

Tableau 6-1 Pourcentage de bien classés

Distances	D_h	D_{hproto}	D_{ch}	D_{vp}	2 étapes
Nombre de bonnes classifications /78	54	60	62	63	66

L'étude des résultats nous donne le même classement que celui de la courbe COR. Les résultats obtenus par la distance D_h sont moins bons que ceux obtenus par D_{ch} qui eux même sont moins bons que ceux obtenus par D_{vp} en une étape et enfin les meilleurs résultats sont

obtenus par la distance D_{vp} en deux étapes. On observe ici des résultats légèrement meilleurs lorsque la distance D_h est utilisée sur des couples (vecteur, prototype), mais D_{vp} et D_{ch} obtiennent de meilleures performances.

Cependant le nombre d'exemples à classer n'étant que de 78, il faut relativiser les résultats qui ne permettent pas de mettre nettement en relief les différences entre les différentes distances. La distance de Hamming permet de classer correctement 70% des exemples. Cela montre la variabilité obtenue entre les exemples d'un même défaut, bien que les données proviennent d'un simulateur. Celui-ci a réussi à générer des situations différentes, proches du réel, ce qui permet de rendre non trivial le problème de diagnostic du système.

6.3.2.1.3. Matrices de confusion

Afin d'étudier plus finement les résultats obtenus par les différentes distances, nous traçons les matrices de confusions présentées dans le chapitre 5. Tous les défauts dont la distance entre le vecteur du défaut à diagnostiquer et le prototype est inférieure à un seuil de détection T_d sont listés comme défauts possibles et présentés à l'opérateur. Afin de comparer les performances des méthodes, le seuil de détection est fixé de tel sorte que pour un nombre X de 78 exemples, le bon diagnostic est présent dans la liste des défauts proposés (la trace de la matrice de confusion est égale à X). On analyse ensuite, pour chaque défaut, la liste des défauts possibles obtenue, via la matrice de confusion. Le réglage de T_d pour la distance D_{vp} , D_{ch} et D_h a été effectué pour $X=77$. Notons que nous utiliserons ici la distance de Hamming avec des prototypes car elle semble donner de meilleurs résultats sur la base du CERN. Les matrices sont représentées sous forme d'images.

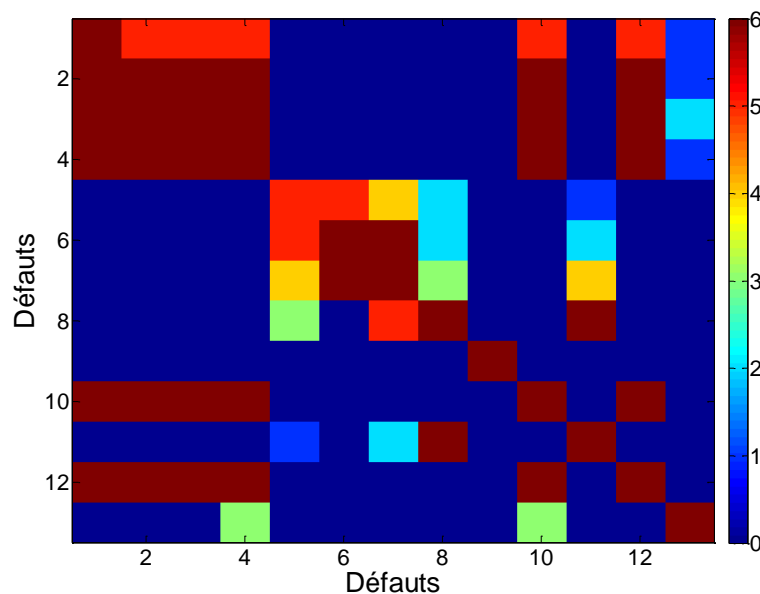


Figure 6-18 Matrice de confusion D_h

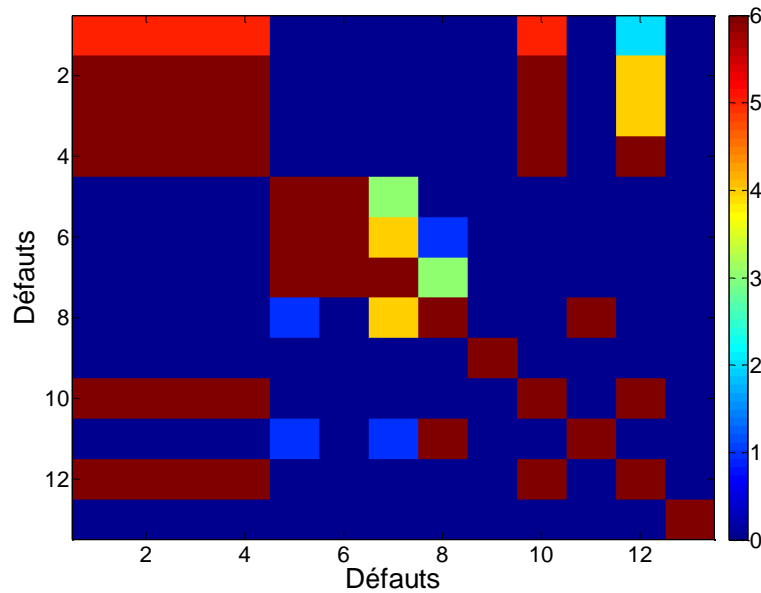


Figure 6-19 Matrice de confusion D_{ch}

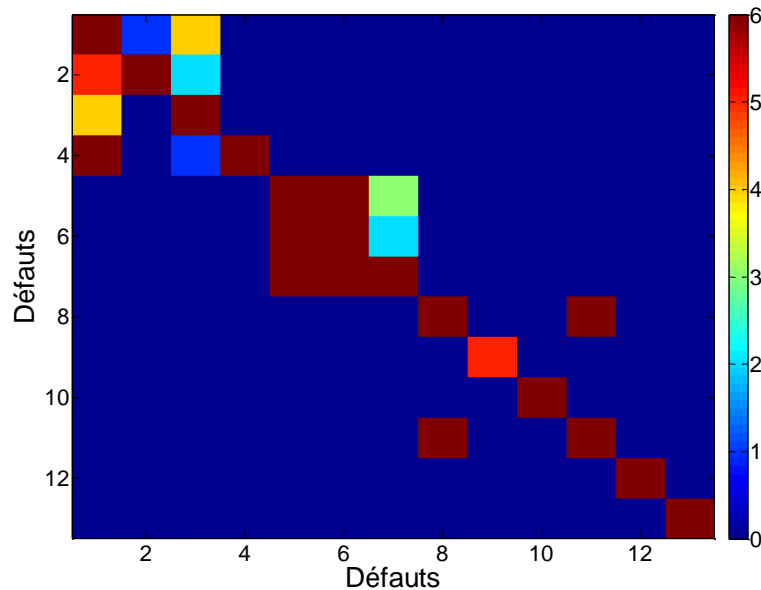


Figure 6-20 Matrice de confusion D_{vp}

Sur ces trois images, les diagonales sont « remplies » à 98% de leur capacité, ce qui est en accord avec le seuil fixé pour la détection.

Une première étude montre qu'une partie des confusions entre défauts, quelle que soit la distance, se fait aux alentours de la diagonale de la matrice. Cela peut se justifier par le fait que les défauts sont numérotés en fonction de leur position dans le système. Deux défauts proches géographiquement ont des numéros proches. Plus deux défauts sont proches géographiquement, plus ils génèrent les mêmes alarmes, et donc plus ils sont difficiles à distinguer. En effet, le système gaz, comme nous l'avons vu au chapitre 3, est une circulation

de gaz en boucle fermée, suivant toujours le même sens et le même parcours. Deux défauts ayant lieu à des endroits proches vont alors avoir tendance à suivre le même chemin que le gaz dans le circuit et donc à déclencher les mêmes alarmes. Le nombre moyen de propositions faites à l'opérateur s'obtient en sommant tous les éléments sur une même ligne et en divisant par 6 (le nombre d'exemples par défaut). Cette information est représentée Figure 6-21, en fonction de chaque défaut. D'une manière générale, le nombre de propositions faites à l'opérateur est de 2.15 en moyenne lorsque D_{vp} est utilisée, avec un maximum de 3 propositions. Pour comparaison, le nombre moyen de propositions obtenu avec la distance D_h est de 6, et de 4.3 avec la distance D_{ch} , le nombre de défauts proposés pouvant monter jusqu'à 6 pour D_{ch} et D_h . En utilisant D_{vp} , on réduit donc le choix de l'opérateur de 13 à au plus 3 solutions possibles.

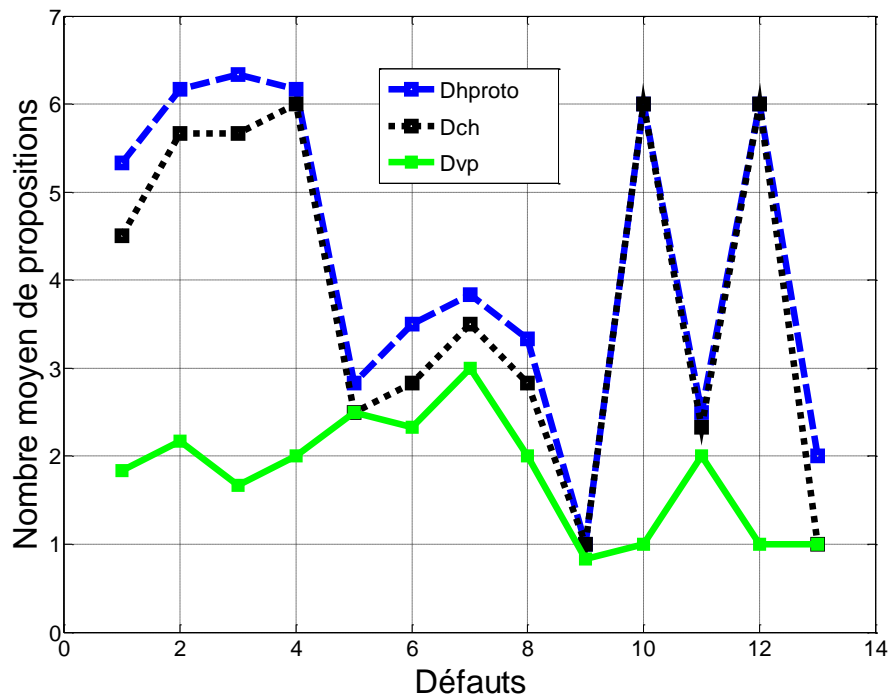


Figure 6-21 Nombre moyen de propositions

L'analyse de la figure 6-21 et des matrices de confusion montre que deux des défauts, le défaut 9 et le 13 ne sont jamais confondus quelle que soit la distance choisie (D_h , D_{ch} , D_{vp}). Etudions les raisons d'un point de vue système de cette facilité à diagnostiquer ces deux défauts.

i. Défaut 9

Le défaut numéro 9 correspond à un arrêt de la pompe du module « Pompe », ceci correspond à un très sérieux défaut pour le système. Il est celui qui affecte le plus l'état du système, en effet il s'agit d'un blocage de la circulation du gaz au cœur de la boucle de circulation (comme le montre Figure 6-22) avec une augmentation très rapide de la pression

qui ne peut pas être compensée par d'autres éléments du système. La gravité du défaut déclenche alors un très grand nombre d'alarmes, dans tous les modules du système, le rendant facile à diagnostiquer pour les trois distances D_h , D_{ch} et D_{vp} .

L'analyse des poids faite au paragraphe **Erreur ! Source du renvoi introuvable.** avait déjà montré un très grand nombre d'alarmes spécifiques au défaut 9.

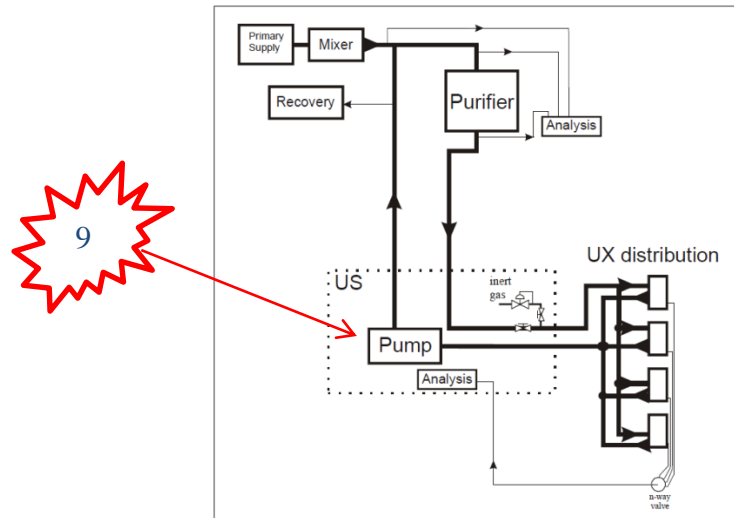


Figure 6-22 localisation défaut 9

ii. Défaut 13

Intéressons-nous maintenant au défaut 13 qui est relativement bien diagnostiqué quel que soit la distance. Ce défaut reste localisé sur une petite partie du système gaz qui ne se situe pas sur le circuit principal de la boucle de circulation du gaz. Le défaut 13 correspond à un bulleur cassé, qui provoque une fuite de gaz au niveau du rack numéro 6 des chambres des détecteurs (voir sur la Figure 6-23 et Figure 6-24):

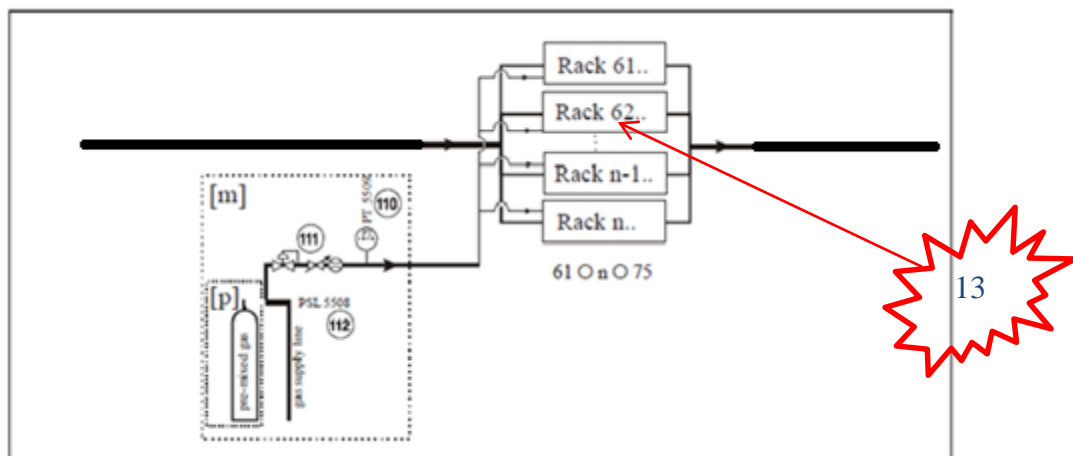
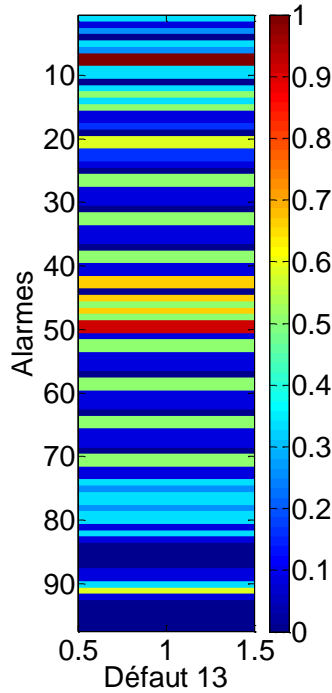


Figure 6-23 localisation défaut (module distribution) 13-1


 Figure 6-25 Poids w_j^i du défaut 13

6.3.3. Apport de la distance D_{vp}

Pour mieux comprendre l'apport de la distance D_{vp} , on étudie l'influence des poids des défauts qui sont correctement diagnostiqués par D_{vp} et pas par D_{ch} ou D_h . Pour ce faire, on s'intéressera aux groupes de défaut confondu lorsque D_{ch} et D_h sont utilisées et pas lorsque D_{vp} est utilisée.

i. Groupe (5,6,7,8)

Intéressons-nous d'abord au groupe (5,6,7,8). Les défauts 5, 6 et 7 sont largement confondus avec le défaut 8 avec D_h et D_{ch} mais jamais lorsque D_{vp} est utilisée (une analyse d'un point de vue système des raisons de la confusion entre ces défauts est proposée au chapitre 8). Nous analyserons ici les poids calculés sur la base réduite aux défauts 5,6, 7 et 8 présentés Figure 6-26 pour faciliter l'analyse, les alarmes prenant systématiquement la valeur 0 pour ces 4 défauts ont été exclus de la figure). Les poids prennent des valeurs faibles pour les alarmes 5, 6 et 7, ce qui montre qu'il n'existe pas d'alarmes permettant de discriminer ces trois défauts, d'où leur confusion systématique. En effet, les alarmes déterministes prennent les mêmes valeurs pour les trois défauts. La distance D_{vp} permet par contre de parfaitement différencier ces 3 défauts du défaut 8, alors que la distance de D_h et de D_{ch} ont tendance à les confondre. La Figure 6-26 met effectivement en évidence plusieurs alarmes spécifiques au défaut 8 par rapport à 5, 6 et 7.

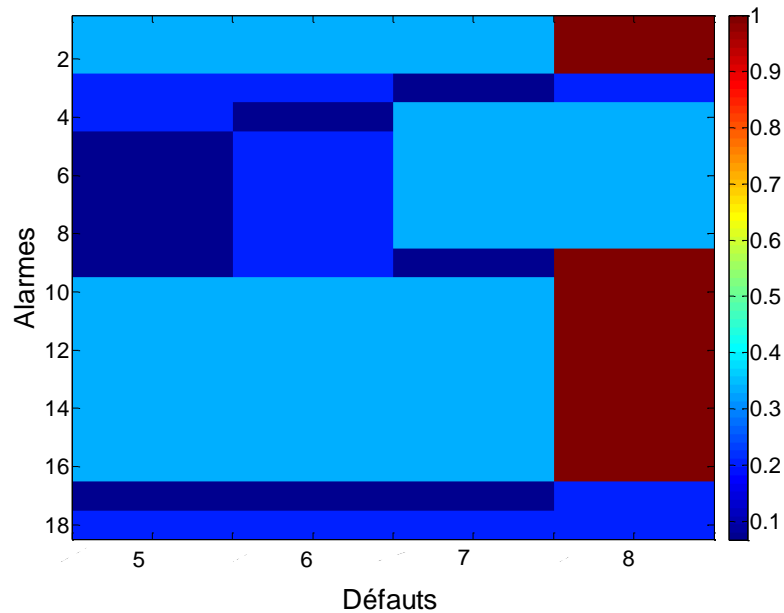


Figure 6-26 Poids défaut 5,6,7 et 8

ii. Groupe (1, 2, 3, 4, 10 et 12)

Les matrices de confusions font ressortir que les groupes de défauts (1, 2, 3, 4, 10 et 12) sont confondus par D_{ch} et D_h mais très rarement voir jamais par D_{vp} . Le nombre moyen de propositions faites à l'opérateur est ainsi drastiquement réduite lorsque D_{vp} est utilisé pour ce groupe de défaut, il passe de 6 avec D_h et D_{ch} (les défauts 1, 2, 3, 4, 10 et 12 sont systématiquement proposés) à 2 voire 1 pour les défauts 10 et 12.

Regardons d'abord d'un point de vue système pourquoi ces défauts sont confondus. Les défauts (1, 2, 3 et 4) sont tous localisés dans le module « Mixeur » (Figure 6-27) du système gaz. Les défauts 1 et 2 correspondent à des fuites dans les bouteilles fournissant du gaz pur au système alors que les défauts 3 et 4 correspondent à un problème de régulation dans les vannes MFC, mène à un blocage de la distribution du gaz. Même si la nature des défauts est différente, il n'est pas surprenant qu'ils déclenchent des listes d'alarmes similaires. Leur conséquence est identique sur le système : ils génèrent une pénurie de gaz pur pour le circuit principal, les gaz purs n'étant plus distribués, ce qui aboutit à un arrêt du module mixer.

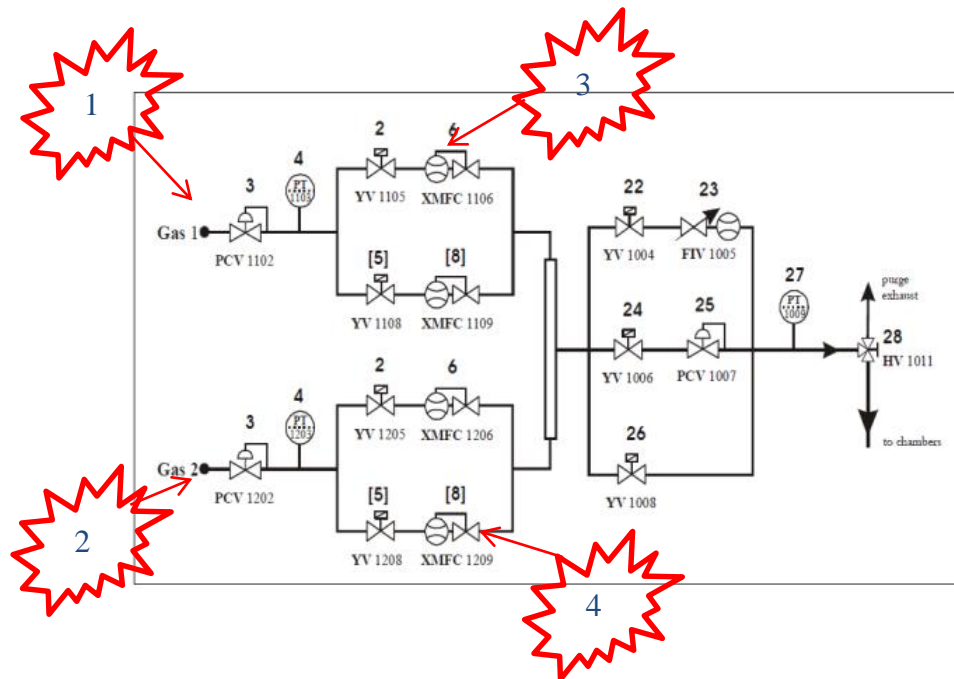


Figure 6-27 Localisation des défauts 1,2,3 et 4

Les défauts 10 et 12 correspondent également à des fuites de gaz, l'une est due à une fuite dans un tuyau au niveau de la pompe et l'autre correspond à une fuite due au bulleur cassé. Ces fuites déclenchent les mêmes listes d'alarmes car elles affectent en premier lieu toutes les deux le module mixeur. Ce module est le premier à s'arrêter car il est le plus « fragile » et le plus susceptible de se mettre en défaut. Cet exemple est une belle illustration de la difficulté du diagnostic du système gaz. Ici les premières alarmes déclenchées ne sont pas celles liées directement au défaut, ce qui montre que les premières alarmes déclenchées ne sont pas forcément les plus représentatives.

Regardons maintenant la composition des vecteurs d'alarmes de ces défauts (Figure 6-28). Les défauts sont représentés sous forme de matrice et affichés sous forme d'image. Chaque colonne de la matrice est un vecteur d'alarmes. Une alarme de valeur égale à 1 est affichée en noir une alarme de valeur égale à 0, en blanc. On accole les 6 exemples dont on dispose. Les 6 premières colonnes représentent donc les 6 exemples du défaut, les 6 suivantes (numérotés de 7 à 12) etc... Pour des soucis de clarté, les alarmes qui ne sont déclenchées dans aucun des différents exemples ne sont pas représentées.

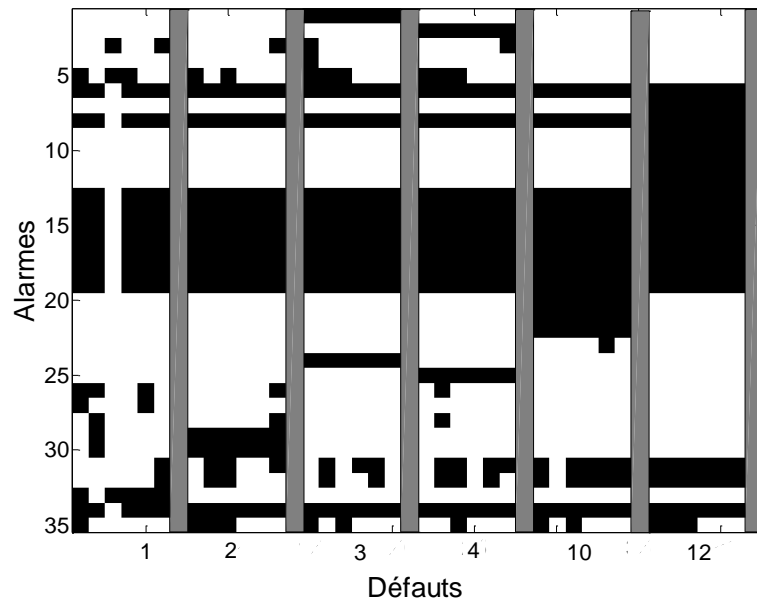


Figure 6-28 Représentation des vecteurs de défauts 1,2,3,4,10,12

L'observation des alarmes met en évidence la grande similarité des défauts entre eux. De nombreuses alarmes sont communes à tous les défauts. Pour mieux visualiser et comparer ces défauts les uns par rapport aux autres, nous représentons les matrices des poids ($2 * \alpha_j^i - 1$), représentées Figure 6-29. Celles-ci montre que, hormis pour le défaut 1, la majorité des alarmes prend systématiquement la même valeur (soit 0 soit 1) sur l'occurrence du défaut.

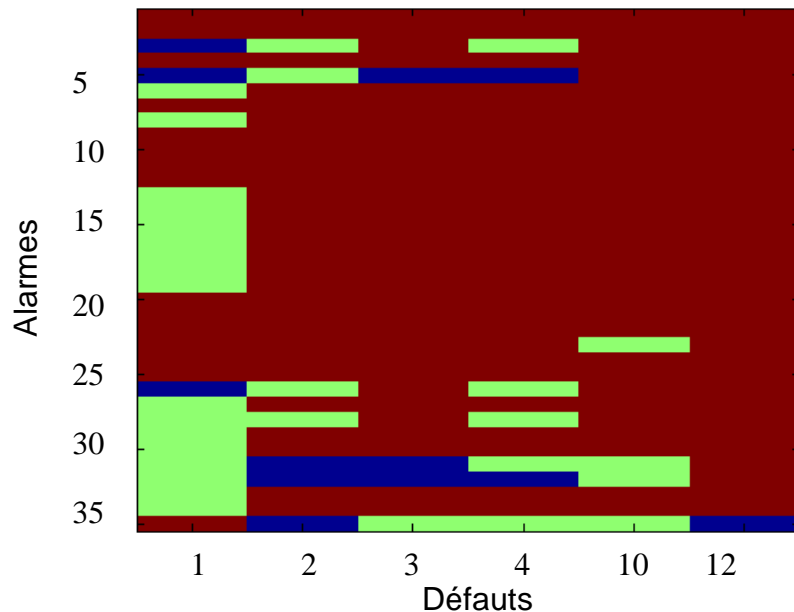


Figure 6-29 Poids ($2 * \alpha_j^i - 1$) calculées pour les défauts 1,2,3,4,10 et 12

L'analyse des poids $(2 * \alpha_j^i - 1) * (1 - \beta_j^i)$ calculés quand l'ensemble des défauts est restreint à (1, 2, 3, 4, 10 et 12) montre qu'il n'existe que très peu d'alarmes typiques d'un défaut. Ceci rend le diagnostic difficile voire impossible lorsque les distances D_h et D_{ch} sont utilisées. Cependant, le défaut 12 contient 5 alarmes spécifiques et le 10 en contient 3, ce qui permet à D_{vp} de parfaitement les isoler du reste par rapport aux défauts (1, 2, 3, 4). Ceci permet aussi une meilleure discrimination des défauts (1, 2, 3, 4) même si elle n'est pas parfaite du fait du faible nombre d'alarmes discriminantes.

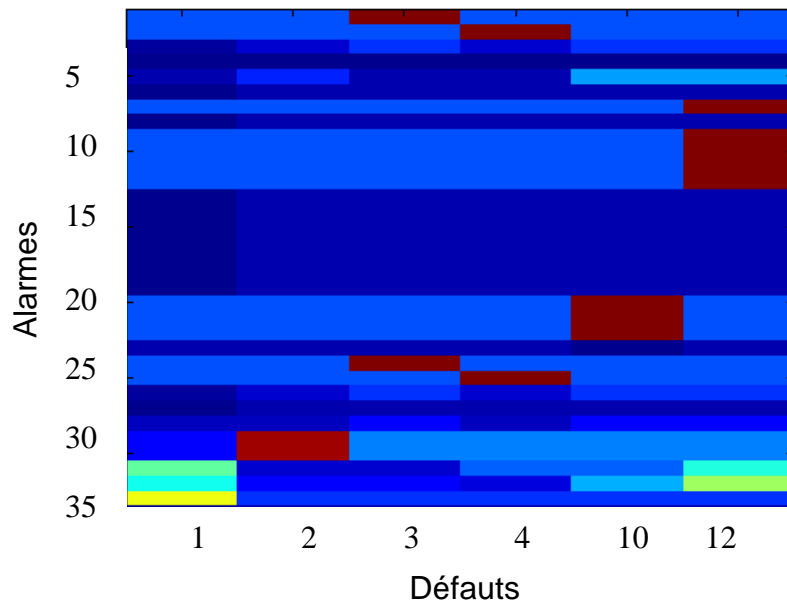


Figure 6-30 Poids complets des défauts 1, 2, 3, 4, 5 et 6

6.3.3.2. Rejet de défauts non contenus dans la base d'apprentissage

Afin d'étudier la capacité de rejet de la distance D_{vp} , on trace la courbe COR en rejet, en faisant varier le seuil T_r de détection. Chacun des 78 exemples est analysé de la manière suivante :

- le prototype du défaut est inclus dans la base d'apprentissage. Si la liste des défauts dont la distance au prototype est inférieure au seuil T_r n'est pas vide, l'exemple est compté comme une bonne détection.
- le prototype du défaut est exclu de la base d'apprentissage. Si la liste des défauts dont la distance au prototype est inférieure au seuil T_r n'est pas vide, l'exemple est compté comme une fausse alarme.

On calcule ensuite, pour l'ensemble des 78 exemples, le pourcentage de bonnes détections et le pourcentage de fausses alarmes.

Les performances obtenues sont relativement médiocres. Ainsi, si l'on choisit le seuil de détection T_r de manière à obtenir 90% de bonnes détections (seulement 10% des exemples

dont le prototype est contenu dans la base d'apprentissage sont détectés comme défaut inconnu et donc non classés), le taux de fausses alarmes est de 40%. Ainsi, près de la moitié des exemples de défauts inconnus sont classés suivant un défaut de la base d'apprentissage. Ces performances ne sont néanmoins pas surprenantes, compte tenu de l'analyse qui a été faite de la base de données. Nous avons montré précédemment qu'il était relativement difficile de parfaitement isoler un défaut en utilisant un seuil sur la distance. De nombreux défauts ont une signature relativement similaire. Les défauts 1, 2, 3 et 4 sont fréquemment confondus, ainsi que les défauts 5, 6 et 7. Ceci est d'autant plus vrai que la distance utilisée pour le rejet n'inclue pas les termes associés aux β , qui met l'accent sur les alarmes spécifiques à un défaut par rapport à d'autres. La distance D_{ch} obtient des résultats équivalents, ce qui n'est pas surprenant puisque l'information utilisée, centrée sur le caractère déterministe de chaque alarme, est similaire. Les performances sont encore plus dégradées quand la distance de Hamming est utilisée.

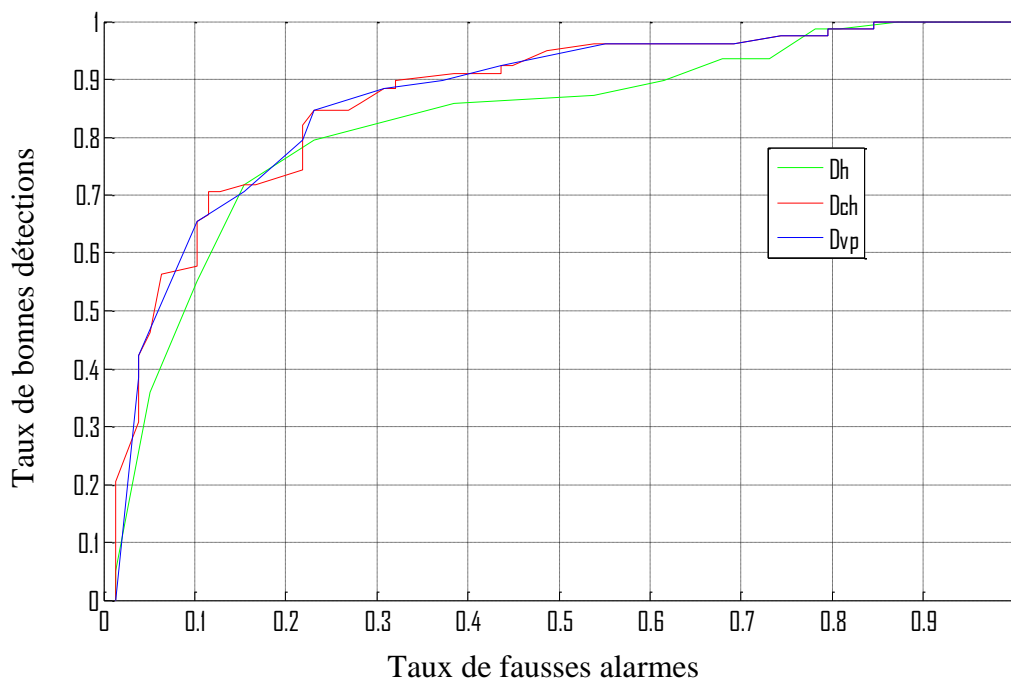


Figure 6-31 Courbe COR rejet

6.4. Évolution du diagnostic au cours du temps

On analyse maintenant le pourcentage d'exemples bien classés en fonction du temps. On a considéré jusqu'à maintenant que le diagnostic était effectué une fois que le flot d'alarmes s'était stabilisé. Nous allons maintenant effectuer le diagnostic chaque fois que de nouvelles alarmes apparaissent dans la liste d'alarmes. Pour se faire, on trace le pourcentage de bien classés parmi les 78 exemples en fonction du temps. La classification est réalisée par la méthode du plus proche voisin. Le diagnostic est réalisé à partir de l'information disponible

depuis le déclenchement de la première alarme jusqu'à l'instant t . Les prototypes et les poids sont ainsi recalculés au cours du temps.

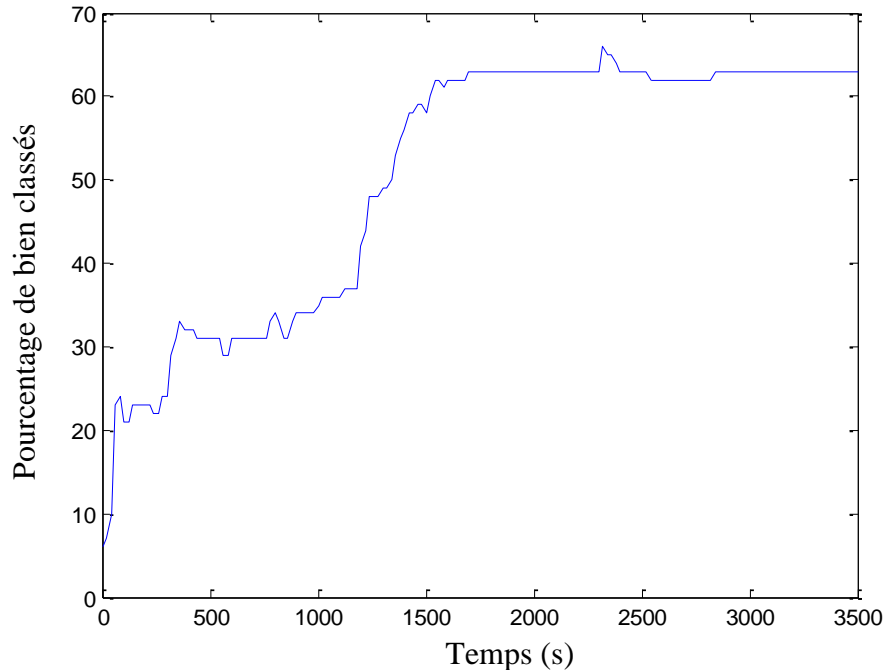


Figure 6-32 Pourcentage de bien classés en fonction du temps

L'observation de la courbe nous permet de dégager 3 phases de temps. La première entre 0 et 1200 secondes où le pourcentage de bien classés augmente très faiblement, suivi d'une deuxième phase où le pourcentage augmente très rapidement pour se stabiliser lors de la dernière phase. La première phase correspond à une période où le nombre d'alarmes déclenchées n'est pas suffisant pour différencier les défauts présents. Après 1200 secondes, la quantité d'information devient suffisante pour différencier les défauts entre eux. Enfin, après un certain temps, les alarmes déclenchées n'aident plus au diagnostic et constituent du bruit. Elles ne sont plus significatives car trop éloignées dans le temps de l'origine du défaut.

6.5. Conclusion

Dans ce chapitre, les performances de la méthode développée au chapitre 4 sont évaluées sur deux jeux de données distincts :

Une base de données constituée de données artificielles et une base constituée des vecteurs d'alarmes générés par le simulateur du système gaz du CERN présenté au chapitre 3. La distance D_{vp} est comparée à deux méthodes de la littérature, la distance de Hamming et la distance proposée par Charkaoui et al. Les outils d'analyse de performances, présentés au chapitre 5, mettent en évidence les performances sur chacune des deux bases. Sur les deux bases et avec tous les outils de mesure de performances (courbe COR, pourcentage de bien

classés par la méthode du plus proche voisin et matrice de confusion) D_{vp} montre de meilleurs résultats que les deux autres distances utilisées pour la comparaison. Les résultats sur la base issue du simulateur montrent que la méthodologie peut apporter une aide à l'opérateur avec un taux de bonnes détections de l'ordre de 95% pour un taux de fausses alarmes inférieur à 5%. Une étude des défauts non confondus et confondus par les méthodes est aussi présentée dans ce chapitre. On s'intéresse d'abord aux défauts confondus en étudiant ces défauts d'un point de vue du procédé. Cette étude montre que la principale raison de la confusion entre les défauts réside dans la proximité géographique de celle-ci. Puis, on propose une explication des raisons de l'erreur de diagnostic faite par les distances à partir des listes d'alarmes. Elle provient du très grand nombre d'alarmes communes entre les défauts qui pose problème lors du diagnostic utilisant D_h et D_{ch} . D_{vp} , à l'aide de l'expression de ses poids en fonction de β_j^i , permet de passer outre cette difficulté et présente de très bons résultats sur la base données du simulateur. Le nombre moyen de propositions de défauts possibles fait à l'opérateur est drastiquement réduit. Enfin, l'influence du temps est évaluée à la fin de ce chapitre afin de déterminer le temps moyen nécessaire à un diagnostic.

7. Diagnostic à l'aide de comparaison de séquences d'alarmes

Dans ce chapitre, nous présenterons d'abord le formalisme adopté pour représenter les listes d'alarmes sous forme de séquences symboliques. Après une présentation des différents outils permettant la manipulation des séquences, les méthodologies d'extraction de prototypes séquentiels ainsi que les méthodes de comparaison des séquences d'alarmes entre elles seront explicitées. Enfin, les méthodes d'aide à l'opérateur et de diagnostic utilisant les prototypes séquentiels seront présentées.

7.1. Formalisme

7.1.1. Présentation des données

Comme précédemment, N est le nombre d'alarmes pouvant être déclenchées par le système de contrôle. L'ensemble de ces alarmes est représenté par $E = \{a_1, \dots, a_j, \dots, a_N\}$ avec « a_j » l'alarme numérotée j . Une alarme peut alors avoir deux états : actif ou inactif. Une alarme sera à l'état actif si les conditions nécessaires à sa présence sont remplies (la valeur d'une variable est au-dessus du seuil limite par exemple). Inversement, une alarme sera dite inactive si les conditions nécessaires à sa présence ne sont pas satisfaites. A titre d'illustration, la Figure 7-1 ci-dessous présente une alarme se déclenchant lorsqu'une variable physique franchit un seuil haut. L'évolution de la valeur de la variable physique en fonction du temps est représentée en noir sur la figure, le seuil de déclenchement (Seuil_h) de l'alarme en rouge et enfin les différentes phases (actif et inactif) sont représentées sur le graphique.

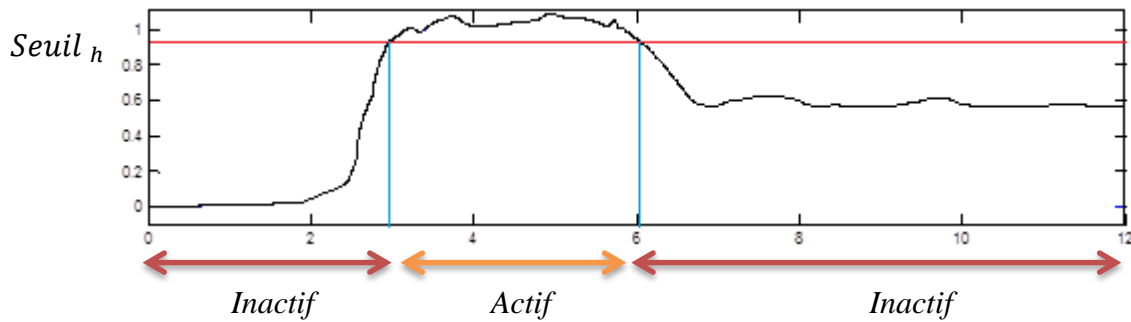


Figure 7-1 Alarme avec déclenchement par franchissement de seuil

On définira pour chaque alarme deux phases de transitions d'un état à l'autre. On notera a_j^+ le passage de l'état inactif à l'état actif et a_j^- le passage de l'état actif à inactif de l'alarme a_j . Soit $\Omega = \{a_1^+, a_1^-, \dots, a_j^+, a_j^-, \dots, a_N^+, a_N^-\}$ l'ensemble des transitions possibles dans le système de contrôle. On appellera alphabet l'ensemble Ω des transitions, formé de $2 \times N$ symboles. La Figure 7-1 reprend l'exemple précédent en traçant le chronogramme de l'alarme associée à cette variable. Les phases de transitions représentent les éléments a_j^+ (encadré en orange dans la figure) et a_j^- (encadré en vert dans la figure).

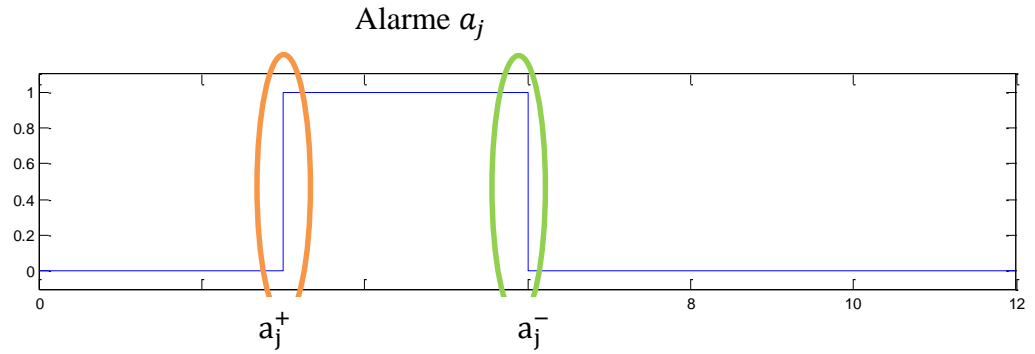


Figure 7-2 Chronogramme de l'alarme a_j

Par la suite, on considérera les éléments de l'ensemble des transitions Ω comme des éléments indépendants que nous traiterons indifféremment. Chaque transition, de l'état actif à inactif, ou de inactif à actif, sera donc considérée comme un symbole distinct de l'alphabet Ω .

7.1.2. Définition d'une séquence d'alarmes

Soit $S = \langle s_1, \dots, s_q, \dots, s_l \rangle$ un ensemble ordonné de l symboles avec $s_q \in \Omega \forall q$. On appellera séquence cet ensemble. Chaque élément de la séquence S respecte la relation suivante $t_{q-1} \leq t_q$ où t_q est l'instant d'apparition s_q dans la liste d'alarmes. Chaque symbole représente le passage à l'état actif ou inactif d'une alarme.

Du fait du temps de cycle de l'automate, ainsi que de la structure interne du programme automate, il est possible que plusieurs alarmes soient déclenchées dans un même temps de cycle et qu'ainsi elles soient horodatées du même instant. Il arrive donc assez souvent que les éléments de la liste d'alarmes arrivent par groupe et non pas l'un après l'autre. Dans ce cas-là, on respectera toujours le même l'ordonnancement que celui du système de supervision. Un même groupe d'alarmes est toujours ordonnée de la même manière par le système de supervision, il suffit alors de respecter cet ordre lors de la construction de la séquence d'alarmes.

7.1.3. Exemple

Afin d'illustrer le formalisme, on propose de reprendre l'exemple donné au chapitre 4. Soit la liste d'alarmes déclenchées pour une faute f^i , où l'on considère un système pouvant générer 5 alarmes, soit un ensemble de transitions $\Omega = \{a_1^+, a_1^-, a_2^+, a_2^-, a_3^+, a_3^-, a_4^+, a_4^-, a_5^+, a_5^-\}$ composé de 10 symboles.

Ainsi si le système de supervision délivre la liste ordonnée d'alarmes suivante :

Tableau 7-1 Liste d'alarmes

Heure de déclenchement	Nom de l'alarme	Etat
00 :00 :01	a_1	Activée

00 :00 :02	a_4	Activée
00 :00 :02	a_1	Désactivée
00 :00 :03	a_2	Activée

La séquence d'alarmes sera :

$$S = \langle a_1^+, a_4^+, a_1^-, a_2^+ \rangle$$

7.1.4. Notations

Dans la suite de ce chapitre, on reprendra les notations du chapitre 4 que l'on rappelle ici:

Soit F l'ensemble des m défauts différents contenus dans la base de cas.

$$F = \{f^1, \dots, f^i, \dots, f^m\} \quad 7-1$$

Avec

f^i le défaut numéro i

La liste d'alarmes de l'exemple $q \in \{1 \dots k\}$ du défaut $i \in \{1 \dots m\}$ est représentée par la séquence suivante :

$$S^{i,q} = \langle s_1^{i,q}, \dots, s_p^{i,q}, \dots, s_n^{i,q} \rangle \quad 7-2$$

Soit A l'ensemble d'apprentissage :

$$A = \{(S^{1,1}, f^1), (S^{1,2}, f^1), \dots, (S^{2,1}, f^2), (S^{2,2}, f^2), \dots, (S^{m,1}, f^m), (S^{m,k}, f^m)\} \quad 7-3$$

La taille du jeu de données d'apprentissage est donc de $k \times m$.

7.1.5. Problématique et algorithmes de recherche de séquences

Notre problématique de diagnostic devient donc un problème de comparaison de séquences symboliques. Une séquence d'alarmes S générée par un défaut inconnu doit être comparée à l'ensemble des séquences contenues dans la base de cas afin de déterminer quel défaut est à l'origine de la séquence. L'approche utilisée est celle du pattern matching. De manière similaire à la méthode proposée quand les listes d'alarmes sont représentées sous forme vectorielle, un prototype séquentiel est construit à partir des exemples d'un même défaut. Ensuite, une mesure de similarité entre la séquence à diagnostiquer et chaque prototype de défaut est proposée afin de sélectionner le défaut dont le prototype est le plus similaire.

Il existe deux approches possibles pour comparer des séquences symboliques. On peut comparer deux séquences en recherchant une concordance exacte entre elles. L'algorithme le plus connu dans ce domaine est sans doute l'algorithme de Knuth-Morris-Pratt (Knuth, et al., 1987) utilisé dans la recherche de mots dans un texte. La deuxième approche consiste à comparer deux séquences en déterminant un degré de similarité entre elles. Les séquences peuvent être similaires sans être rigoureusement les mêmes. Ce sont les approches utilisées en bio-informatique pour comparer des séquences de gènes. Dans notre problématique de diagnostic de défauts, nous avons vu au chapitre 1 que, sous l'influence de nombreux paramètres (environnement extérieur, mode de fonctionnement, système physique non déterministe ...), les listes d'alarmes générées par le système de contrôle sont souvent très différentes même lorsqu'elles sont déclenchées par un même défaut. Ainsi, dans la suite de ce chapitre, la méthodologie de diagnostic proposée pour comparer des séquences d'alarmes utilisera des algorithmes bio-informatiques.

7.2. Similarité entre deux séquences symboliques – l'algorithme de Needleman and Wunsch

7.2.1. Alignement de deux séquences

En bio-informatique, aligner deux séquences consiste à réarranger des chaînes d'ADN, ARN ou de protéines dans le but de trouver les zones de fortes similarités entre ces séquences. L'alignement se fait par l'ajout de «gaps» dans chacune des deux séquences afin d'avoir un maximum de concordance entre les éléments des séquences se trouvant aux mêmes positions.

Voici un exemple d'alignement de deux séquences :

Soit S^1 et S^2 deux séquences d'ADN ayant comme alphabet $\Omega = \langle A, T, C, G \rangle$

$$\begin{aligned} S^1 &= \langle A, T, C, C, G, T \rangle \\ S^2 &= \langle A, T, G, C, C, G, A \rangle \end{aligned} \quad 7-4$$

L'alignement de ces deux séquences sera représenté par deux nouvelles séquences notées S^{a^1} et S^{a^2} :

$$\begin{aligned} S_2^{a^1} &= \langle A, T, -, C, C, G, T \rangle \\ S_1^{a^2} &= \langle A, T, G, C, C, G, A \rangle \end{aligned} \quad 7-5$$

On notera la séquence résultante de l'alignement de S^1 par rapport à S^2 , $S_2^{a^1}$ et inversement on notera la séquence résultant de l'alignement de S^2 par rapport à S^1 , $S_1^{a^2}$. On étendra cette notation à l'alignement de deux séquences (S^{r_1}, S^{r_2}) , avec $S_{r_2}^{a_{r_1}}$ la résultante de l'alignement S^{r_1} avec S^{r_2} , et $S_{r_1}^{a_{r_2}}$ la résultante de l'alignement S^{r_2} avec S^{r_1} . Cette notation sera également étendue à l'alignement multiple que nous décrirons dans le paragraphe 7.3.1.

En introduisant un « gap » à la position 2 de la séquence S^1 , on maximise la concordance entre les deux séquences pour chaque position de la séquence. Dans cette configuration, l'ajout du « gap » sur la séquence 1 sera considéré comme étant la résultante d'une suppression d'un élément pour la séquence 1 et d'une insertion pour la séquence 2. On appelle longueur de l'insertion/suppression le nombre de « gaps » successifs nécessaires à l'alignement, dans notre exemple la longueur est de 1.

7.2.2. Similarité entre deux séquences symboliques

La manipulation des séquences afin de les aligner ainsi que le principe de concordance, repose sur l'idée de similarité entre deux séquences. Aligner deux séquences, c'est chercher à maximiser la similarité entre celles-ci. Il est donc important avant tout alignement de définir une mesure de similarité de deux séquences que l'on cherchera à maximiser lors de l'alignement de ces deux séquences. Pour cela, il est tout d'abord nécessaire de définir la similarité entre deux symboles. Soit un alphabet constitué de $2N$ symboles, $\Omega = \{s_1, \dots, s_{2N}\}$, la similarité entre symboles est définie à l'aide d'une matrice $2N \times 2N$, notée C , où chaque élément $C(i, j)$ fixe la similarité entre le symbole s_i et le symbole s_j . De même, le coût d'un gap, ω , fixe la pénalité associée à l'introduction d'un gap dans la séquence. Soit S_2^{a1} et S_1^{a2} deux séquences alignées de taille l , on définit la similarité entre les deux séquences noté sim_{NW} :

$$S_2^{a1} = \langle s_1^{a1}, \dots, s_p^{a1}, \dots, s_l^{a1} \rangle \quad 7-6$$

$$S_1^{a2} = \langle s_1^{a2}, \dots, s_p^{a2}, \dots, s_l^{a2} \rangle \quad 7-7$$

$$sim_{NW}(S_2^{a1}, S_1^{a2}) = \sum_{p=1}^{p=l} \sigma_p \quad 7-8$$

Avec la fonction σ_p définie de la manière qui suit :

Si s_p^{a1} et s_p^{a2} sont deux éléments de l'alphabet alors $\sigma_p = C(s_p^{a1}, s_p^{a2})$ où $C(s_p^{a1}, s_p^{a2})$ est la valeur du coût de substitution de l'élément s_p^{a1} par l'élément s_p^{a2} . Si s_p^{a1} ou s_p^{a2} correspond à un gap, alors $\sigma_p = \omega$.

7.2.3. Algorithme de Needleman et Wunsch - alignement de deux séquences

Il existe deux types d'alignements de séquences, l'alignement local et l'alignement global. L'alignement local recherche des régions de forte similarité dans les séquences alors que l'alignement global cherche à maximiser la similarité entre les deux séquences dans leur globalité. Il existe pour chaque type d'alignement une méthode de résolution optimale. (Needleman & Wunsch, 1970) ont proposé une méthode de résolution optimale pour l'alignement global de deux séquences utilisant une matrice de coût. T.F. Smith et M.S.

Waterman (Smith & Waterman, 1981) ont proposé une version modifiée de l'algorithme de Needleman et Wunsch afin de réaliser un alignement local.

Dans le cas d'analyse de séquences d'alarmes, il est intéressant de considérer la séquence dans sa totalité. Le risque lorsque des similarités locales sont recherchées est de trouver certaines sous-séquences d'alarmes qui correspondent à des réactions du système de contrôle alors qu'il paraît plus judicieux de prendre en compte la propagation totale du défaut sur le système plutôt que ces effets dans chaque module. On s'intéressera donc à l'algorithme de Needleman and Wunsch afin d'aligner les séquences. Dans la suite, il est présenté pour l'alignement de deux séquences S^1 et S^2 de longueurs respectives l_1 et l_2 .

$$\begin{aligned} S^1 &= \langle s_1^1, \dots, s_p^1, \dots, s_{l_1}^1 \rangle \\ S^2 &= \langle s_1^2, \dots, s_q^2, \dots, s_{l_2}^2 \rangle \end{aligned} \quad 7-9$$

7.2.3.1. Principe de l'algorithme

L'objectif de l'alignement de deux séquences est de maximiser la similarité à l'aide de l'ajout de gaps. L'algorithme de Needleman et Wunsch propose de réaliser cet alignement optimal à l'aide du principe de récursivité qui a été démontré par (Smith & Waterman, 1981). L'idée générale est que l'alignement entre la sous séquence S^1 composée des éléments de S^1 depuis s_1^1 jusqu'à s_p^1 et la sous-séquence S^2 composée des éléments de S^2 depuis s_1^2 jusqu'à s_q^2 peut être déterminé à partir de l'alignement de la sous séquences S^1 depuis s_1^1 à s_{p-1}^1 et la sous-séquence S^2 depuis s_1^2 jusqu'à s_{q-1}^2 .

Une matrice de dimension $(l_1 + 1, l_2 + 1)$, que l'on appelle matrice de comparaison, notée *Comp*, est ainsi construite de manière récursive. Les éléments de S^1 sont représentés en lignes et ceux de S^2 en colonnes. Les éléments *Comp* donne les valeurs de similarités de tous les alignements possibles.

La construction de la matrice est réalisée de la manière suivante. On commence l'indexation à 0 afin de réserver la première ligne et la première colonne à la phase d'initialisation.

On initialise cette matrice de la manière qui suit :

$$Comp(p, 0) = p * \omega \text{ et } Comp(0, q) = q * \omega \quad 7-10$$

Chaque case de la matrice est alors calculée de la façon suivante :

$$\begin{aligned} Comp(p, q) &= \max\{Comp(p-1, q-1) \\ &\quad + C(s_p^1, s_q^2), Comp(p, q-1) \\ &\quad + \omega, Comp(p-1, q) + \omega\} \end{aligned} \quad 7-11$$

Pour chaque case (p,q), l'algorithme choisit la solution optimale à partir de trois cas possibles :

Si la valeur maximale est atteinte pour :

$$Comp(p-1, q-1) + C(s_q^1, s_q^2), \quad 7-12$$

Alors l'algorithme matche les symboles s_q^1, s_q^2

Si la valeur maximale est atteinte pour :

$$Comp(p, q-1) + \omega \quad 7-13$$

Alors l'algorithme ajoute un gap à S^1

Si la valeur maximale est atteinte pour :

$$Comp(p-1, q) + \omega \quad 7-14$$

Alors l'algorithme ajoute un gap à S^2

Une fois la matrice *Comp* complètement remplie, il est alors possible de déterminer l'alignement optimal de ces deux séquences en suivant le chemin optimal depuis $Comp(l_1 + 1, l_2 + 1)$, jusqu'à $Comp(1,1)$. Le chemin optimal est obtenu en passant d'une case $Comp(p,q)$ à la case adjacente dont la valeur est optimale. Les cases adjacentes à $Comp(p,q)$ sont $Comp(p-1,q-1)$, $Comp(p-1,q)$ et $Comp(p,q-1)$.

Le tableau ci-dessous illustre un exemple à partir de deux séquences définies à partir d'un alphabet $\Omega = \{A, T, C, G\}$

$$\begin{aligned} S^1 &= \langle A, T, G, T \rangle \\ S^2 &= \langle T, C, G \rangle \end{aligned} \quad 7-15$$

On posera les paramètres suivant :

$$\omega = -0.5 \quad 7-16$$

$$C = \begin{matrix} & \begin{matrix} A & T & C & G \end{matrix} \\ \begin{matrix} A \\ T \\ C \\ G \end{matrix} & \begin{pmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{pmatrix} \end{matrix} \quad 7-17$$

La matrice de comparaison est donc la suivante :

Tableau 7-2 Exemple Needleman and Wunsch

		T	C	G
	<u>0</u>	-0.5	-1	-1.5
A	<u>0.5</u>	-1	-1.5	-2
T	-1	<u>0.5</u>	<u>0</u>	-0.5
G	-1.5	0	-0.5	<u>1</u>
T	-2	-0.5	-1	<u>-0.5</u>

Par exemple, la valeur de la case $\text{comp}(2,2)$ est obtenue en sélectionnant le maximum parmi ces trois éléments :

- $\text{Comp}(p-1, q-1) + C(s_q^1, s_q^2) = 0 - 1 = -1$
 - $\text{Comp}(p, q-1) + \omega = -0.5 - 0.5 = -1$
 - $\text{Comp}(p-1, q) + \omega = -0.5 - 0.5 = -1$
- 7-18

Ici, les trois valeurs sont égales la valeur de la case est égale à -1.

Pour construire l'alignement optimal, on démarre de la dernière case pour remonter à la case (1,1). Le chemin à suivre est tracé à l'aide de flèches bleues :

On commence par sélectionner la case qui se trouve en (4,4), ce qui signifie que l'on se « déplace » vers le haut sur une même colonne. Ceci correspond à une insertion pour S^1 , il s'agit donc d'un gap pour la séquence S^2 . On a donc :

—
T

On se déplace ensuite sur la diagonale, il s'agit d'une concordance d'où :

G —
G T

On se déplace sur la gauche sur la même ligne, d'où une insertion pour la séquence S^2 . On ajoute donc un gap à S^1 :

C G —

– G T

On se déplace encore sur la diagonale. Il s'agit d'une concordance, d'où :

T C G –

T – G T

Et enfin on se déplace sur une même colonne vers le haut d'où l'alignement final :

– T C G –

A T – G T

7.3. *Extraction de prototypes séquentiels*

Nous disposons de k séquences symboliques représentant un même défaut. La construction d'un prototype de défaut va nécessiter dans un premier temps l'alignement des k séquences. La séquence minimale correspondant au prototype sera ensuite extraite à partir des k alignements.

7.3.1. *Alignement multiple*

Un alignement multiple consiste à aligner plus de deux séquences entre elles en procédant au même type de réarrangements que celui fait avec deux séquences, c'est-à-dire par l'ajout de « gaps ». L'algorithme utilisé par Needleman and Wunsch peut aisément être généralisé pour k séquences. Il suffit d'étendre la matrice bidimensionnelle à une matrice de k -dimensionnelle. Cependant, l'algorithme possède alors une complexité exponentielle avec l'augmentation de k . Il a été démontré que le problème était NP-Difficile.

Il n'existe, à l'heure actuelle, aucune méthode de résolution optimale en un temps raisonnable. Cependant, différentes méthodes non-optimales mais donnant des résultats satisfaisants sont utilisées. La littérature fourmille de méthodes proposant des alignements multiple (Corpet, 1988) (Higgins & Sharp, 1988) (Edgar, 2004), elles reposent pour la plupart sur un principe d'alignement progressif introduit par (Feng & Doolittle, 1987). La méthode utilise des heuristiques et procède par la construction progressive de l'alignement. Les méthodes de résolution progressive suivent deux phases :

Construction d'un arbre appelé arbre guide où les relations entre les différentes séquences sont représentées.

Puis les manipulations sur les séquences sont exécutées en suivant l'arbre guide afin de réaliser les alignements.

En bio-informatique, l'arbre guide est une représentation de l'évolution de certaine séquence d'ADN et possède donc un sens biologique intéressant. Les méthodes utilisées dans la littérature reposent pour la plupart sur des propriétés propres aux séquences étudiées et sur

des caractéristiques propres aux séquences. Par exemple, on suppose que certaines séquences sont les ancêtres d'autres séquences. Dans notre cas, nous n'avons a priori pas de relation de la sorte entre les séquences. Nous proposons alors une classification hiérarchique en construisant l'alignement en même temps que la construction de l'arbre à l'aide d'une classification hiérarchique ascendante.

7.3.1.1. Classification hiérarchique ascendante

L'objectif de la classification hiérarchique ascendante (CHA) est de regrouper k séquences de façon successive jusqu'à l'obtention d'un groupe unique contenant k séquences alignées.

La procédure de regroupement et d'alignement progressif de k séquences est décrite ci-dessous :

On suppose que l'on dispose d'une mesure de similarité entre deux séquences

Phase d'initialisation : On calcule les C_k^2 similarités entre les k séquences à aligner. A partir de ces C_k^2 similarités, on construit un tableau de k lignes et k colonnes résumant les similarités entre séquences.

Phase 1 : On sélectionne les deux séquences les plus similaires. Ces deux séquences sont alors alignées et composent le premier groupe que l'on nommera nœud. Les gaps introduit sur chaque séquence seront remplacés par un élément neutre noté X. Cet élément neutre aura un coût nul qu'il soit aligné avec un gap, lui-même ou tout autre élément de l'alphabet.

Phase 2 : On calcule la similarité entre le nœud nouvellement formé et les $(k-2)$ séquences restantes. La similarité entre une séquence et un nœud est définie par la similarité maximale entre la séquence et chaque séquence alignée du nœud (les séquences alignées sont le résultat des séquences originelles avant alignement auquel l'élément X est ajouté à la place des gaps résultant de leur alignement).

Phase 3 : On met à jour le tableau de similarités en éliminant les deux lignes et colonnes correspondant aux deux séquences sélectionnées en phase 1 et en ajoutant une ligne et une colonne portant la similarité du nœud formé avec les $k - 2$ séquences restantes.

Phase 4 : On sélectionne les deux éléments les plus similaires. Il peut s'agir de (séquence/séquence), ou de (séquence/nœud).

Phase 5 : On aligne les deux éléments sélectionnés. L'alignement entre deux séquences se fait de façon classique à l'aide de l'algorithme de Needleman and Wunsch. L'alignement entre un nœud et une séquence s'effectue de la manière suivante: la séquence est comparée avec les séquences alignées du nœud. Puis, la séquence est alignée avec la séquence du nœud ayant la plus grande similarité. Une fois l'alignement réalisé entre les

deux séquences, l'alignement est répercuté à toutes les autres séquences des nœuds à l'aide de la règle « once a gap always a gap » c'est-à-dire que lorsqu'un gap est introduit, il est transmis à toutes les autres séquences du nœud.

La suite de l'algorithme consiste à itérer phases 2, 3, 4 et 5 jusqu'à ne former qu'un seul groupe :

La similarité utilisée pour calculer les différentes similarités lors de la classification hiérarchique est la similarité calculée à l'aide de l'algorithme de Needleman and Wunsch normée en fonction de la longueur des séquences. Ceci permet d'obtenir une similarité insensible à la longueur des séquences et ainsi pouvoir comparer les similarités entre elles.

La similarité normée est définie par :

$$sim_{NWnorm}(S^{a^1}_2, S^{a^2}_1) = \frac{\sum_{p=1}^{p=l} \sigma_p}{l} \quad 7-19$$

Avec l la longueur des séquences $S^{a^1}_2$ et $S^{a^2}_1$

7.3.1.2. Exemple alignement multiple

Afin de mieux comprendre le principe d'alignement multiple, on se propose d'étudier l'exemple suivant :

Soit 4 Séquences : S^1, S^2, S^3, S^4 défini par un alphabet {A,T,C,G}.

$$\begin{aligned} S^1 &= \langle A, T, C, C, G, T \rangle \\ S^2 &= \langle A, T, G, C, C, G, A \rangle \\ S^3 &= \langle T, C, G, C, T, C, G, T \rangle \\ S^4 &= \langle T, C, G, G, C, G, T \rangle \end{aligned} \quad 7-20$$

On fixera les paramètres de la manière qui suit :

$$\omega = -0.5 \quad 7-21$$

et

$$C = \begin{matrix} & \begin{matrix} A & T & C & G \end{matrix} \\ \begin{matrix} A \\ T \\ C \\ G \end{matrix} & \begin{pmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{pmatrix} \end{matrix} \quad 7-22$$

D'où le tableau de similarité suivant :

Tableau 7-3 Similarités 1

	S^1	S^2	S^3	S^4
S^1	X			
S^2	0.5	X		
S^3	0.33	0.28	X	
S^4	0.44	0.125	0.56	X

On choisit la plus grande similarité dans le tableau :

Les deux séquences les plus proches sont les séquences S^3 et S^4 . Elles constituent donc le premier nœud. On aligne les deux séquences afin de créer le nœud noté $S^{a3,4}$ composé des deux séquences alignées:

$$S^{a3,4} = \begin{cases} S_4^{a3} = < T, C, G, C, T, C, G, T > \\ S_3^{a4} = < T, C, G, -, G, C, G, T > \end{cases} \quad 7-23$$

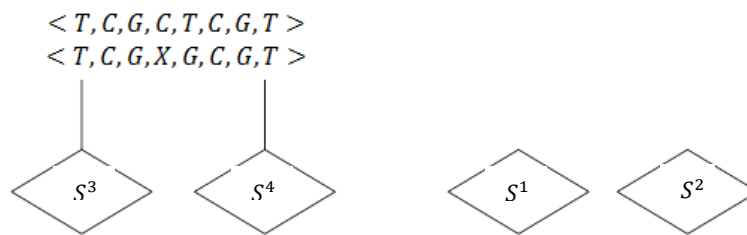


Figure 7-3 Classification hiérarchique étape 1

On calcule maintenant les distances entre les deux séquences restantes, puis celle entre le nœud et les séquences S^1 et S^2 .

La matrice de similarité mise à jour est présentée ci-dessous:

Tableau 7-4 Similarités 2

	S^1	S^2	S^3
S^1	X		
S^2	0.5	X	
$S^{a3,4}$	0.38	0.28	X

La plus grande similarité est obtenue pour l'alignement des séquences 1 et 2. Figure 7-4 Classification hiérarchique étape 2

On aligne les deux séquences afin de créer le nœud de ces deux séquences noté $S^{a1,2}$ et composé des deux séquences alignées:

$$S^{a^1}_2 = \langle A, T, -, C, C, G, T \rangle$$

$$S^{a^2}_1 = \langle A, T, G, C, C, G, A \rangle$$

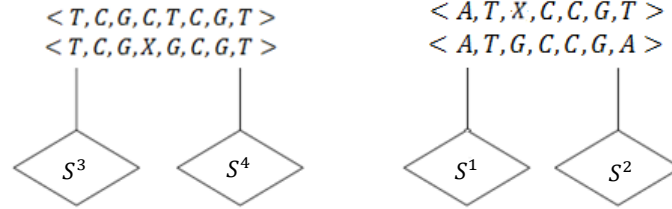


Figure 7-4 Classification hiérarchique étape 1

Il ne reste plus qu'une seule possibilité pour l'alignement final. Il faut aligner les deux groupes restants. Le calcul de similarité entre les séquences alignées des deux groupes donne une similarité maximum pour l'alignement de $S^{a^2}_1$ avec $S^{a^3}_4$ avec un score égal à 0.38. On alignera $S^{a^1}_2$ avec $S^{a^3}_4$ puis on propagera les gaps aux autres membres du groupe.

Pour créer le dernier groupe et donc l'alignement de toutes les séquences, on applique la règle «once a gap always a gap».

Soit l'alignement entre $S^{a^1}_2$ et $S^{a^3}_4$:

$$\begin{aligned} S^{a^{1,2}}_{3,4} &= \langle A, T, -, X, C, -, C, G, T \rangle \\ S^{a^{3,4}}_{1,2} &= \langle -, T, C, G, C, T, C, G, T \rangle \end{aligned} \quad 7-24$$

On étend ici la notation des alignements avec $S^{a^{1,2,3,4}}$ la séquence résultant de l'alignement de $S^{a^{1,2}}$ avec $S^{a^{3,4}}$.

L'alignement final après la propagation des gaps Alignement :

$$S^{a^{1,2,3,4}} = \begin{cases} \langle A, T, -, -, C, C, -, G, T \rangle \\ \langle A, T, -, G, C, -, C, G, A \rangle \\ \langle -, T, C, G, C, T, C, G, T \rangle \\ \langle -, T, C, G, -, G, C, G, T \rangle \end{cases} \quad 7-25$$

L'alignement final est donc le suivant. On a propagé les gaps des alignements entre l'alignement $S^{a^3}_4$ et $S^{a^1}_2$ aux deux autres alignements $S^{a^2}_1$ et $S^{a^4}_3$.

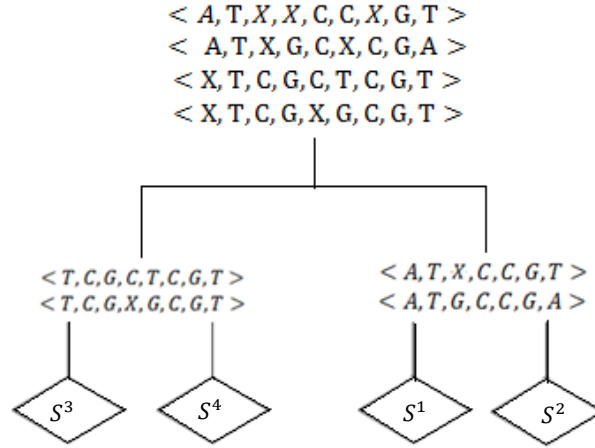


Figure 7-5 Classification hiérarchique étape 3

7.3.2. Extraction de la séquence minimale

7.3.2.1. Principe

Pour construire le prototype du défaut, on procède à l'alignement des k exemples du défaut i à l'aide de l'algorithme d'alignement multiple présenté dans le paragraphe 7.2. On notera $S^{a(1,...,k)^i}$ l'alignement multiple des k exemples du défaut i .

L'alignement multiple $S^{a(1,...,k)^i}$ peut être considéré comme une matrice de dimension k (nombre de séquences alignées) et l (longueur des séquences composant l'alignement multiple final). Cette matrice est notée M^{al^i} .

Soit $M_{r,q}^{al^i}$ l'élément (r, q) de cette matrice qui correspond à l'élément à la position q de la séquence de l'alignement (l'exemple numéro r du défaut i).

On notera $fr(x, q)$ la fréquence de l'élément $x \in \Omega$ à la position q de l'alignement calculé de la manière qui suit :

$$fr(x, q) = \sum_{r=1}^{r=k} \frac{\delta(M_{r,q}^{al^i}, x)}{k} \quad 7-26$$

$$\text{Avec } \delta(M_{r,q}^{al^i}, x) = 1 \text{ si } M_{r,q}^{al^i} = x \text{ et } \delta(M_{r,q}^{al^i}, x) = 0 \text{ sinon}$$

Le prototype est construit alors à partir des éléments x tel que $fr(x, q) \geq \lambda$, on note l_p le nombre de ces éléments x . Le prototype séquentiel sera alors de longueur l_p . Il est constitué des éléments x tel que $fr(x, q) \geq \lambda$, présentés suivant leur ordre d'apparition dans la matrice $M_{r,q}^{al^i}$.

On notera P^i le prototype de longueur l_p :

$$P^i = \langle p_1^i, \dots, p_u^i, \dots, p_{lp}^i \rangle \quad 7-27$$

Avec $1 \leq lp < l$

L'alignement multiple permet de trouver les concordances entre les différents éléments des séquences alignées. L'extraction de prototypes repose sur le principe de la fréquence d'apparition, comme lors de la construction des prototypes vectoriels. Mais, ici, en plus de la fréquence d'apparition, une condition nécessaire est ajoutée pour que l'alarme appartienne au prototype. Il faut que l'ordre entre elles soit respecté dans les différentes séquences de liste d'alarmes composant le défaut. On construit ainsi des prototypes avec une notion d'ordre d'apparition des alarmes, ce qui nous permet de disposer d'une information supplémentaire pour le diagnostic.

Le paramètre λ est un paramètre de réglage permettant de définir un seuil d'acceptation de l'élément à la position q du prototype séquentiel. λ est compris entre 0.5 et 1. Une valeur supérieure à 0.5 permet d'éviter d'avoir plusieurs solutions possibles pour une position q . En effet, pour un alignement multiple de k séquences à chaque position q , si plus de la moitié des séquences possèdent le même élément appartenant à Ω , alors il n'aura pas d'autre élément possible pour la position q . Prendre un λ inférieur à 0.5, c'est prendre le risque d'avoir un conflit entre deux éléments de Ω pour la position q . Une valeur élevée de λ permet un meilleur filtrage des alarmes qui pourrait être considérées comme du bruit, en ne gardant que celles apparaissant fréquemment dans les séquences avec le même ordre d'apparition. La construction des prototypes séquentiels s'apparente à celle des prototypes vectoriels. On conserve les alarmes apparaissant sur une majorité d'exemples suivant le même ordre d'apparition par rapport aux autres alarmes.

7.3.2.2. Exemple

Soit l'alignement:

$$S^{a1,2,3,4} = \begin{cases} \langle A, T, -, -, C, C, -, G, T \rangle \\ \langle A, T, -, G, C, -, C, G, A \rangle \\ \langle -, T, C, G, C, T, C, G, T \rangle \\ \langle -, T, C, G, -, G, C, G, T \rangle \end{cases} \quad 7-28$$

Représenté sous la forme matricielle suivante :

Tableau 7-5 Représentation de l'alignement multiple sous forme de tableau 1

A	T	-	-	C	C	-	G	T
A	T	-	G	C	-	C	G	A
-	T	C	G	C	T	C	G	T
-	T	C	G	-	G	C	G	T

Si l'on reprend l'exemple précédent le prototype séquentiel est le suivant, en imposant $\lambda = 0.5$:

Tableau 7-6 Représentation de l'alignement multiple sous forme de tableau 2

A	T	-	-	C	C	-	G	T
A	T	-	G	C	-	C	G	A
-	T	C	G	C	T	C	G	T
-	T	C	G	-	G	C	G	T

D'où le prototype suivant $P = \langle T, G, C, C, G, T \rangle$.

7.4. Influence des paramètres ω , C et λ

Lors de l'application de l'extraction de prototypes séquentiels, il existe un certain nombre de paramètres à définir. Ce paragraphe discute des différentes influences de ces paramètres ainsi que de leurs valeurs d'affectation possibles.

7.4.1. Paramètre λ

Le paramètre λ sert à définir la fréquence d'apparition minimale permettant d'accepter une alarme dans le prototype séquentiel final. Il agit d'un paramètre de filtrage d'alarmes. Pour une même valeur de λ , le nombre d'alarmes éliminées est cependant plus important que dans le cas du prototype vectoriel, car la condition sur l'ordre d'apparition contraint plus fortement la sélection. Le nombre d'alarmes présentes dans le prototype séquentiel diminue évidemment avec la valeur de λ . Il faudra choisir avec précaution ce paramètre. Un λ trop faible inclura les alarmes qui bruitent l'information sur le défaut. Au contraire, un λ élevé risque d'exclure des alarmes pertinentes pour le diagnostic. Le paramètre λ peut à priori prendre toutes les valeurs entre 0 et 1. Les cas où $\lambda \leq 0.5$ peuvent amener à un conflit lors de la construction du prototype séquentiel, car à une même position, on pourrait avoir 2 alarmes ou plus. Ce cas correspond à un OU logique. Dans ce cas, on peut décider d'inclure ces alarmes dans le prototype dans n'importe quel ordre. L'alignement ultérieur s'occupera de correctement aligner les alarmes du défaut inconnu au prototype séquentiel, en ajoutant un gap aux alarmes manquantes.

7.4.2. Paramètre ω

Le paramètre ω fixe la pénalité de l'ajout d'un gap dans l'alignement des séquences. Un faible coup de pénalité, qui correspond à une valeur de ω élevée signifie que l'on pénalise peu les ajouts de gaps. Ceci est approprié pour des séquences fortement bruitées, qui nécessitent de nombreuses manipulations et ajout de gaps afin de trouver des concordances entre les séquences. Cependant, il existe un inconvénient à fixer une valeur élevée à ω . En ajoutant un grand nombre de gaps dans les séquences, on risque de mettre en concordance des

alarmes situées à des positions très éloignées entre deux séquences. De ce fait, on générera des concordances artificielles entre des alarmes quand celles-ci ne sont pas forcément représentatives du défaut dont on cherche à extraire le prototype séquentiel. De manière similaire, à trop ajouter de gaps, on risque de rater des concordances pertinentes entre alarmes car la séquence a été trop décalée en amont, notamment lors de l'alignement de séquence multiple.

À l'inverse, un ω faible (correspondant à un coût de gap fort) réduit les manipulations sur les séquences. Si les séquences sont fortement bruitées, avec un grand nombre d'alarmes non significatives intercalées entre les alarmes pertinentes pour le défaut, la difficulté à ajouter des gaps sur ces alarmes se traduira par une faible similarité entre séquences d'un même défaut.

7.4.3. Paramètre C

La matrice de substitution où matrice C permet de fixer le coût de substitution d'une alarme par une autre. Les coûts peuvent à priori prendre n'importe quelle valeur, qu'elle soit négative ou positive. Plus une valeur sera positive, plus on signifie que les deux alarmes sont similaires, ce qui privilégiera leur alignement. Dans le cas de la substitution d'une alarme par elle-même, une valeur élevée va relever son importance dans la définition des défauts. Dans le cas de deux alarmes différentes, une valeur élevée va signifier que les alarmes décrivent une situation similaire. Ainsi, la matrice de coût doit être définie en intégrant la connaissance que l'on a du système ou de la signification des alarmes. Si des alarmes sont connues pour être non pertinentes, on peut leur affecter un faible coup de substitution (une valeur négative de substitution par exemple). Des alarmes ayant des significations relativement identiques (par exemple une alarme associée au franchissement d'un seuil haut et une alarme associée au franchissement d'un seuil très haut) pourront avoir un fort coût de substitution (une valeur de C élevée).

7.5. Méthodes de décision

Les méthodes de diagnostic proposées dans ce chapitre sont les mêmes que celles développées au chapitre 4 pour la méthode vectorielle D_{vp} . Nous avons de façon similaire développée un prototype représentant un défaut. L'algorithme de Needleman and Wunsch fournit une mesure de similarité entre deux séquences symboliques, Sim_{NWnorm} (équation 7-19), ce qui va permettre de comparer la séquence inconnue aux prototypes séquentiels de défauts. Toutes les décisions prises s'appuient donc sur les similarités entre la séquence du défaut à diagnostiquer et les m prototypes de défauts.

7.5.1. Rejet d'une séquence d'alarmes

On procède à un test en rejet de la manière suivante. Si la plus grande similarité obtenue entre les prototypes de défauts de la base et la séquence de défaut inconnu est inférieure à un

seuil de rejet noté T_r , alors le défaut est considéré comme inconnu et n'appartient pas à l'ensemble F des défauts inclus dans la base.

7.5.2. Affectation d'une séquence d'alarmes à un défaut contenu dans la base d'apprentissage

7.5.2.1. Méthode du maximum

Si la séquence a passé le test de rejet et est donc considérée comme associée à un défaut contenu dans la base, le défaut qui lui est affecté est celui dont la valeur de similarité avec le prototype est la plus élevée.

7.5.2.2. Décision par seuillage

On fixe un seuil d'acceptation T_d sur la valeur de similarité et on fournit à l'opérateur la liste des défauts possibles, correspondant à l'ensemble des défauts dont la similarité entre le prototype et la séquence inconnue est supérieure au seuil T_d .

7.6. Conclusion

Dans ce chapitre, nous avons dans un premier temps défini le formalisme de transformation d'une liste d'alarmes en séquence symbolique. Une méthodologie inspirée des méthodes utilisées en bio-informatique et adaptée aux séquences d'alarmes a été proposée afin d'extraire les différents prototypes des défauts contenus dans la base de cas. Celle-ci repose sur l'alignement multiple des séquences représentatives d'un même défaut. L'alignement des séquences permet de dégager un prototype représentant chaque défaut à l'aide d'une séquence unique, dans laquelle se trouvent les alarmes apparaissant dans la majorité des exemples d'un même défaut et respectant le même ordonnancement. Une similarité dérivée de la bio-informatique est utilisée pour comparer la séquence inconnue et les prototypes de défauts. Les méthodes de décision utilisées sont ensuite les mêmes que la représentation vectorielle.

La représentation séquentielle apporte, par rapport à la représentation vectorielle, la notion d'ordre dans les alarmes, ce qui peut constituer un élément supplémentaire de discrimination entre les défauts. En contrepartie, les prototypes ne sont conçus qu'à partir de l'information provenant du même défaut, ils n'utilisent pas d'informations provenant des autres défauts. Plusieurs paramètres de réglages doivent être fixés pour extraire les prototypes et calculer la similarité entre séquences, dont les influences ont été discutées dans ce chapitre.

8. Analyse des performances de la
méthode de diagnostic par
comparaison de séquences

Dans ce chapitre, nous nous proposons d'étudier les performances de la méthode dite séquentielle, en utilisant la similarité Sim_{Nwnorm} (équation 8-1). Pour ce faire, nous observerons d'abord les performances de la méthode sur une base de données artificielle et évolutive. Puis nous analyserons les résultats de diagnostic sur la base de données issue du simulateur du système gaz du CERN. Afin de situer ses performances, la méthode sera comparée avec la méthode vectorielle développée au chapitre 4.

8.1. Analyse de performances sur une base de données artificielle.

8.1.1. Création d'une base artificielle

Pour illustrer le fonctionnement de l'algorithme de diagnostic séquentiel, nous avons construit une base artificielle contenant m défauts et k exemples par défaut. Chaque défaut est construit sous forme de séquence symbolique de la manière qui suit :

- Une partie de la séquence est composée de T alarmes toujours présentes quels que soient les défauts. L'ordre d'apparition de ces alarmes est toujours le même pour des exemples du même défaut mais varie entre les défauts deux à deux. Dans la suite du paragraphe, on notera de façon symbolique une alarme z appartenant à ce groupe : τ_z . Ces alarmes sont appelées alarmes constitutives et forment une séquence dite constitutive.
- Une partie composée de Θ alarmes purement aléatoires, avec une probabilité d'apparition dans une séquence de 0.5, quel que soit le défaut. L'ordre d'apparition est purement aléatoire. Dans la suite du paragraphe, on notera de façon symbolique une alarme z appartenant à ce groupe : θ_z . On appellera ces alarmes, alarmes aléatoires. Elles formeront une séquence aléatoire.
- Une partie composée de Γ alarmes avec une probabilité d'apparition de 0.8 et avec un ordre d'apparition toujours identique. Dans la suite du paragraphe, on notera de façon symbolique une alarme z appartenant à ce groupe : γ_z . Ces alarmes formeront une séquence répétée et seront donc appelées « alarmes répétées »
- Enfin, une partie composée de Ψ alarmes déterministes apparaissant uniquement dans les exemples d'un même défaut avec un ordre d'apparition toujours conservé. Dans la suite du paragraphe, on notera de façon symbolique une alarme z appartenant à ce groupe : ψ_z et on l'appellera une alarme appartenant à ce groupe une alarme unique et leur séquence constitue une séquence unique.

Notons que la position de chaque alarme varie d'une séquence à une autre mais toujours en respectant l'ordre d'apparition qui la caractérise.

Toutes les séquences définies sont des séquences que l'on peut retrouver sur les listes d'alarmes déclenchées par le système de supervision d'un système réel. Nous verrons que ces

types de séquences sont présents dans la base de données du simulateur du système gaz du CERN.

Un exemple de cette base est illustré dans le schéma qui suit à l'aide d'alarmes symboliques. Chaque alarme est nommée selon le groupe auquel elle appartient. Afin de mieux comprendre la construction de cette base, nous proposons de schématiser la construction de séquences de défauts artificielles étape par étape. Nous présentons la construction de deux défauts.

Étape 1 :

Définissons d'abord les alarmes constitutives (dont l'ordre est représentatif du défaut qu'il représente). Considérons un cas où ces alarmes sont au nombre de $T = 6$. Nous représentons deux défauts différents, soit deux séquences constitutives différentes:

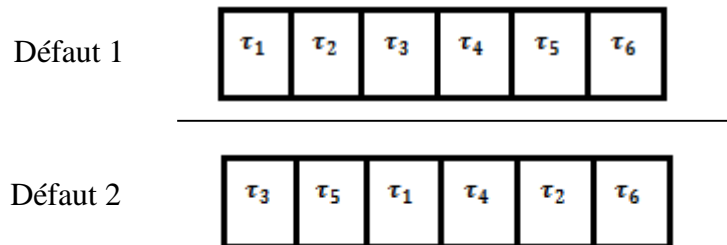
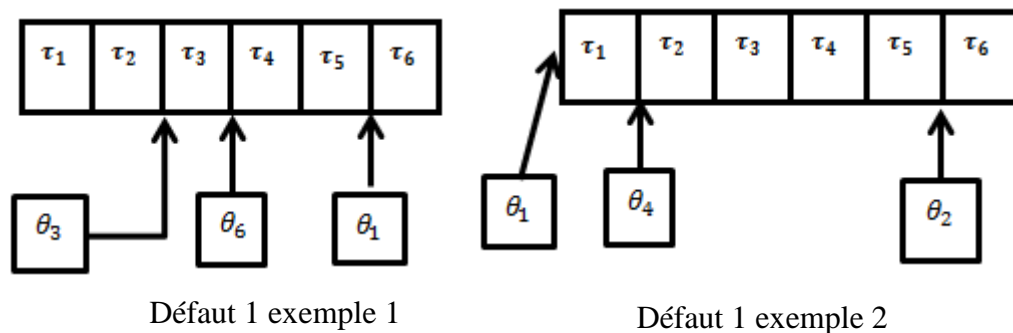


Figure 8-1 Exemple de deux séquences constitutives

On note que, dans les séquences de ces deux défauts, toutes les alarmes sont communes mais leur ordre est différent en fonction du défaut.

Étape 2 :

A partir de ces séquences constitutives, on intègre des alarmes aléatoires au nombre de $\Theta = 6$. On crée deux exemples pour chaque défaut afin d'illustrer la construction des exemples.



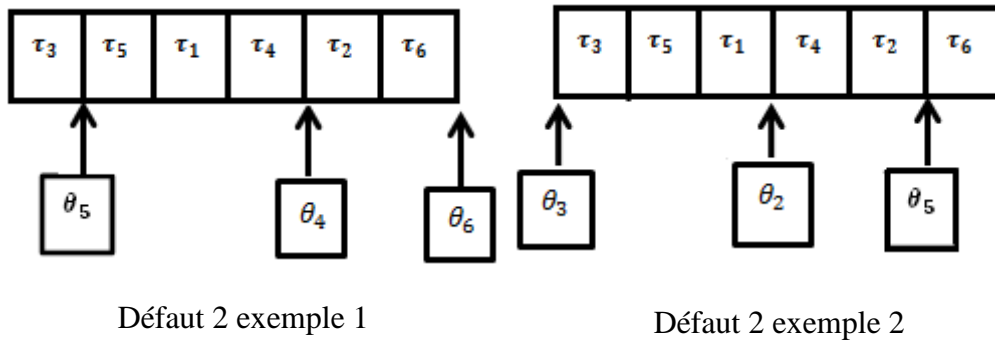


Figure 8-2 Insertion des alarmes aléatoires dans les séquences constitutives

Le schéma montre l'insertion des alarmes dont l'ordre et la probabilité d'apparition sont aléatoires. On remarque que l'ordre d'insertion ne suit pas de règle particulière. La probabilité d'apparition des 6 alarmes est égale à 0.5 sur l'ensemble des exemples.

Étape 3 :

Dans cette étape, la construction des alarmes avec fréquence 0.8 apparaissant toujours dans le même ordre est schématisée (alarmes répétées). On considérera un nombre $\Gamma = 3$. On intègre donc ces alarmes aux deux exemples des défauts précédents.

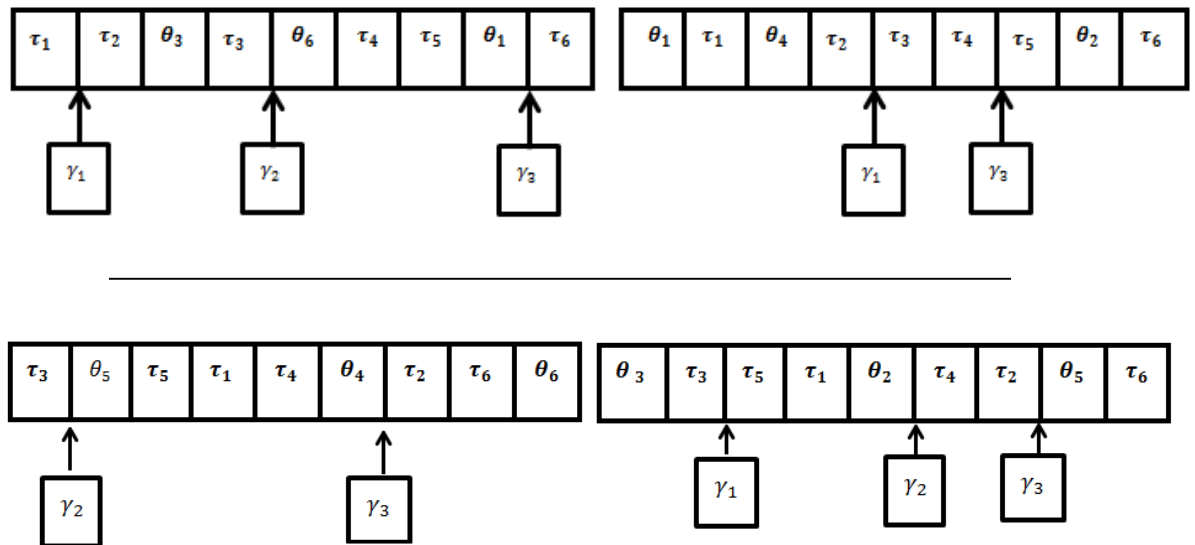


Figure 8-3 Insertion des alarmes répétées

On note ici que l'ordre d'apparition est le suivant : $\gamma_1 \rightarrow \gamma_2 \rightarrow \gamma_3$. Cet ordre est respecté sur les deux exemples des deux défauts. Cependant, leur fréquence d'apparition étant de 0.8, il peut arriver que certaines alarmes n'apparaissent pas sur certains exemples, d'où l'absence de l'alarme γ_2 dans l'exemple 2 du défaut 1 et de γ_1 dans l'exemple 1 du défaut 2. Cependant, l'ordre est quand même respecté : l'alarme γ_1 précède l'alarme γ_3 pour le cas du défaut 1 de l'exemple 2.

Étape 4 :

Dans cette étape, la construction des alarmes uniques est schématisée. On choisit un nombre $\Psi = 2$ pour chaque défaut.

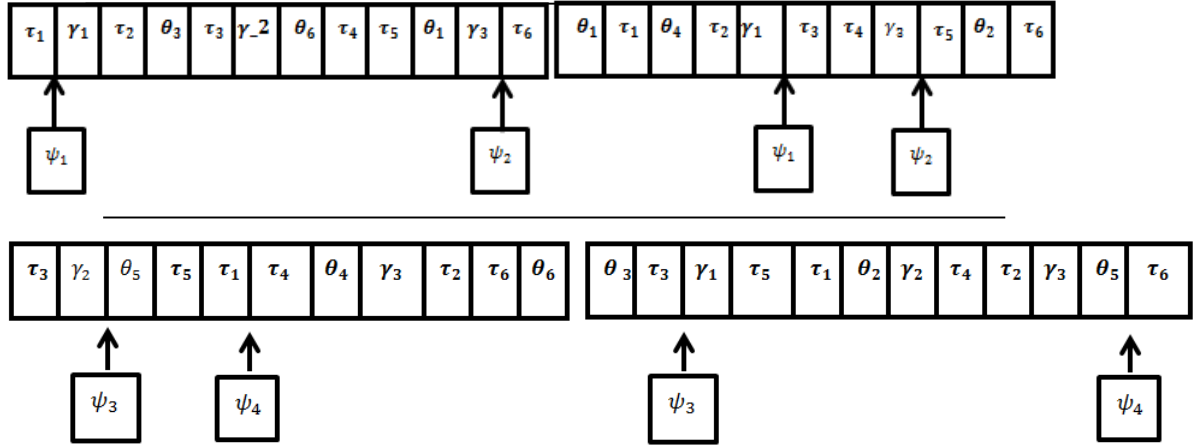


Figure 8-4 Insertion des alarmes uniques

Les alarmes uniques apparaissent uniquement dans les exemples du même défaut (sur le schéma les alarmes ψ_1 et ψ_2 sont propres au défaut 1 et les alarmes ψ_3 et ψ_4 propres au défaut 2). Leur ordre est conservé pour les exemples du même défaut, cependant leur position dans la séquence varie entre les différents exemples.

Un exemple de base avec les paramètres: $m = 13$, $k = 6$, $T = 6$, $\Theta = 40$ et $\Gamma = 5$ $\Psi = 2$ est présenté dans le schéma ci-après en utilisant la représentation sous forme vectorielle des séquences. Pour ce faire on transforme les séquences en vecteurs binaires utilisés au paragraphe 6.

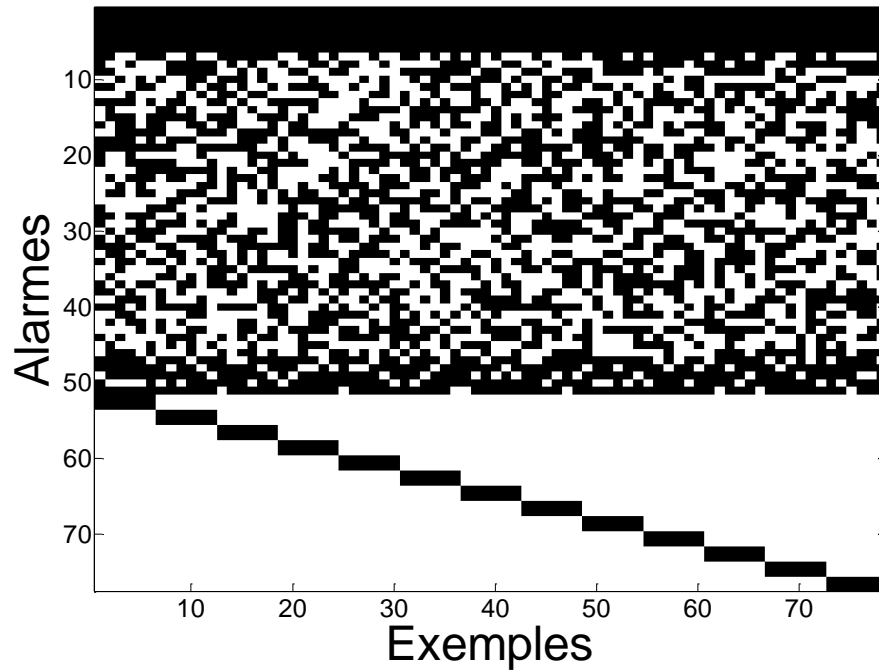


Figure 8-5 Représentation de la base artificielle complète

Les alarmes numérotées de 1 à 6 sont les alarmes présentes quel que soit le défaut avec un ordre différent en fonction du défaut (alarmes constitutives notées τ_z). Les alarmes de 7 à 46 sont les alarmes aléatoires de fréquence d'apparition égale à 0.5 (alarmes notées θ_z), les alarmes de 46 à 50 sont les alarmes aléatoires de fréquence d'apparition égale à 0.8 ayant toujours le même ordre d'apparition quel que soit le défaut ou l'exemple, et enfin, les alarmes de 50 à 76 (notées ψ_z) sont les alarmes uniques apparaissant uniquement dans les exemples du même défaut et toujours dans le même ordre.

8.1.2. Analyses de performances

Nous allons étudier le comportement de la méthode séquentielle en fonction du type de défaut que l'on désire diagnostiquer. Dans ce but, on créera, dans ce paragraphe, différentes bases artificielles en ajoutant à chaque étape un type d'alarmes supplémentaire à la base. On comparera les résultats à chaque étape avec l'algorithme de diagnostic utilisant la distance vectorielle D_{vp} . Ces bases étudient les performances des méthodes sur les différents types de séquences de défaut, sachant qu'une base d'alarmes peut contenir plusieurs types de séquences.

8.1.2.1. Base contenant des séquences constitutives et des séquences aléatoires

Pour ce premier cas, on mesure les performances à l'aide d'une base artificielle définie avec les paramètres suivant $m = 13$ défauts, $k = 6$, $T = 6$, $\Theta = 40$ et $\Gamma = 0$ et $\Psi = 0$. C'est une base contenant 6 alarmes partagées par les 13 défauts mais apparaissant avec des

ordres différents pour deux défauts différents. Cet ordre est cependant conservé pour des exemples du même défaut. Un nombre élevé d'alarmes aléatoires (40) est également présent dans les séquences constituant cette base. La méthode utilisée pour le tracé de la courbe COR est celle présentée dans le chapitre 5 en utilisant la méthode de validation croisée. Les paramètres de la méthode séquentielle pour la suite sont fixés à $\omega = 0$ et $\lambda = 3$. La matrice de substitution C est la suivante (sauf indication contraire) :

$$C = \begin{pmatrix} 1 & -1 & \dots & -1 \\ -1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 \\ -1 & \dots & -1 & 1 \end{pmatrix}$$

i. Résultats avec D_{vp}

La figure ci-dessous montre la courbe COR de la méthode D_{vp} :

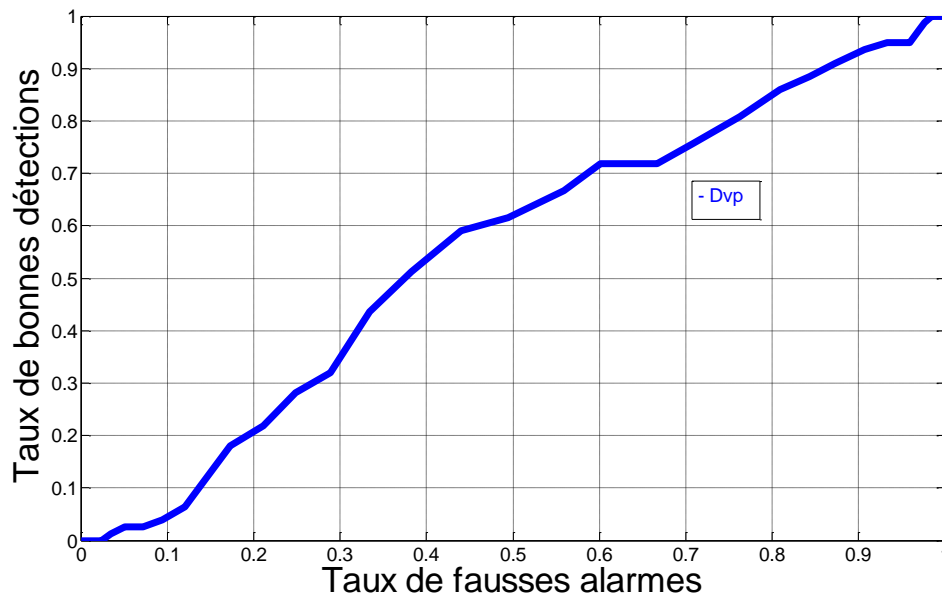


Figure 8-6 Courbe COR méthode vectorielle Dvp

Comme attendu, les résultats de la méthode vectorielle sont mauvais. La courbe COR est proche de la courbe d'équation $y=x$. Cela correspond aux pires performances possibles pour un classifieur binaire, son efficacité est équivalente à une décision purement aléatoire. Il n'est pas surprenant d'avoir de tels résultats, la base de données étant construite de telle sorte qu'il n'existe pas d'alarmes propres à chaque défaut. Les alarmes sont partagées par tous les défauts, quelle que soit la fréquence d'apparition de l'alarme dans les différents exemples.

ii. Résultat Sim_{Nwnorm}

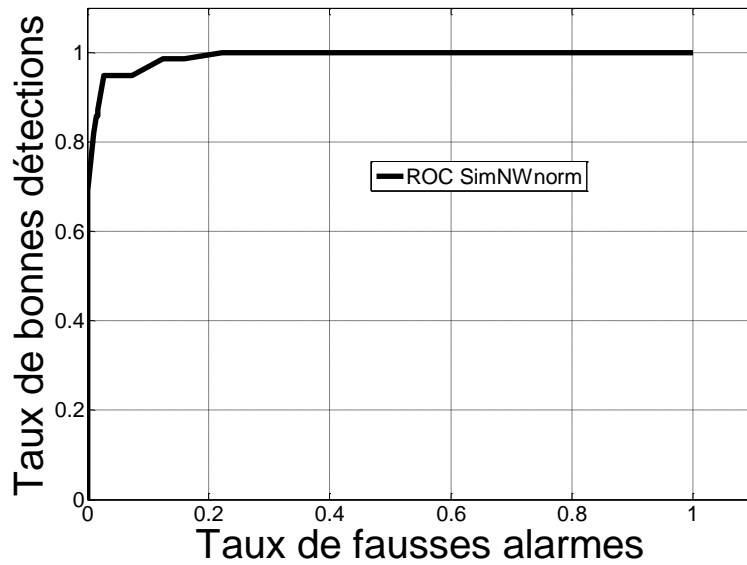


Figure 8-7 Courbe COR méthode Sim_{NWnorm} base artificielle 1

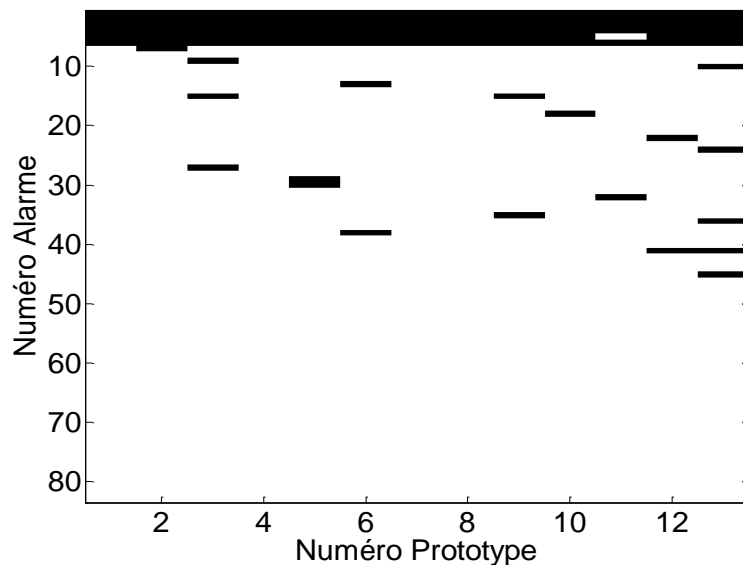


Figure 8-8 Représentation des prototypes séquentiels base artificielle 1

La Figure 8-7 présente la courbe COR de la méthode séquentielle sur la même base afin de pouvoir comparer les résultats.

Les résultats de Sim_{NWnorm} sont bien meilleurs que les résultats utilisant la méthode D_{vp} . La prise en compte de l'ordre d'apparition des alarmes ici est un élément clé de différenciation des défauts entre eux. On observe une courbe COR quasiment parfaite. L'algorithme est capable de déterminer quasiment à coup sûr si le défaut à diagnostiquer et le prototype appartiennent au même défaut (le pourcentage de bonnes détections mesuré est de

l'ordre de 95% pour un pourcentage de fausses alarmes faible). Ces résultats s'expliquent en analysant les prototypes séquentiels extraits (*Figure 8-8*). On peut voir que la plupart des alarmes aléatoires sont filtrées par l'algorithme. Au plus 5 alarmes aléatoires apparaissent dans les prototypes séquentiels des défauts, alors que les alarmes constitutives dont l'ordre est représentatif du défaut (alarmes de 1 à 6) sont toutes incluses dans les prototypes séquentiels à l'exception du défaut 11 auquel il manque une alarme. L'algorithme est donc capable d'extraire les séquences constitutives des défauts même avec un nombre élevé d'alarmes aléatoires (dans notre cas 40, ce qui représente plus de 86% de la séquence). Ces alarmes aléatoires sont filtrées avec succès grâce au paramètre λ . En effet la probabilité d'apparition de ces alarmes étant de 0.5, si la valeur de λ est supérieure à 0.5, elles seront donc pour la grande majorité filtrées. De plus, leur ordre étant complètement aléatoire, il est difficile pour l'algorithme d'alignement multiple d'aligner ces alarmes dans les différentes séquences et donc d'être extraites par l'algorithme d'extraction de prototype séquentiel. A l'inverse, leur présence n'a pas perturbé l'alignement des alarmes constitutives.

8.1.2.2. Base contenant des séquences constitutives, des séquences aléatoires et des séquences répétées

Dans ce qui suit, nous ajoutons à la base précédente des alarmes pseudo aléatoires avec une probabilité d'apparition de 0.8. Celles-ci apparaissent toujours dans le même ordre quel que soit le défaut (séquence répétée). On obtient donc la base suivante : $m = 13, k = 10, T = 6, \Theta = 40, \Gamma = 8$ et $\Psi = 0$ en gardant les mêmes paramètres que précédemment. La *Figure 8-9* montre une représentation vectorielle de cette base de données :

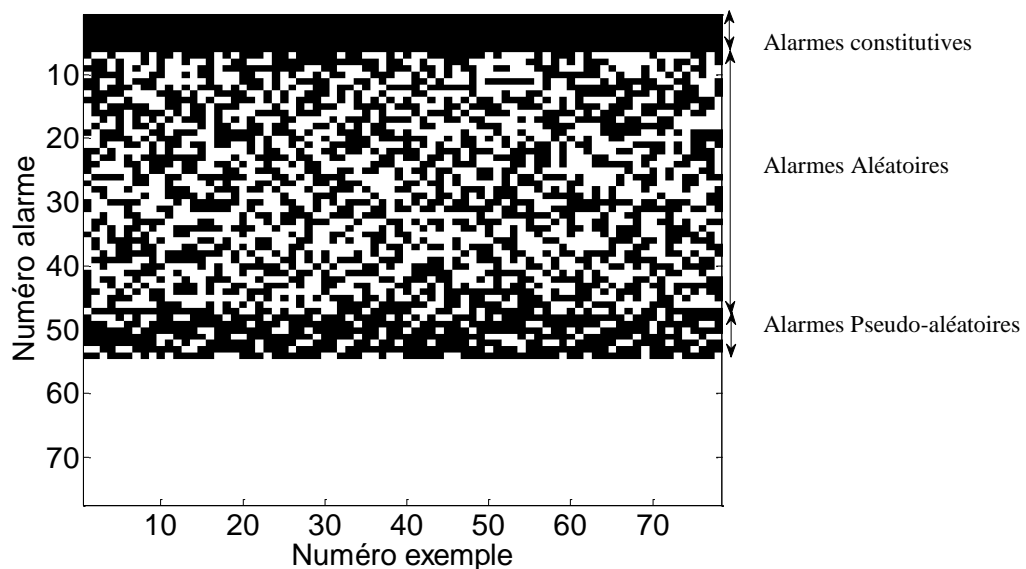


Figure 8-9 Représentation base artificielle 2

i. Résultat D_{vp}

On observe d'abord la courbe COR de la méthode vectorielle utilisant la distance D_{vp} .

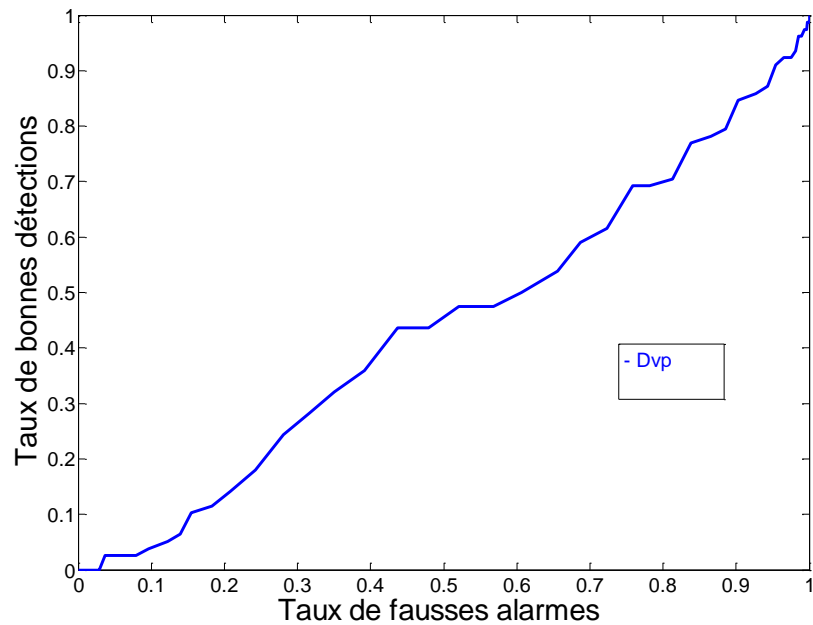


Figure 8-10 Courbe COR Dvp / base artificielle 2

On observe les mêmes résultats que lors de l'analyse avec la base précédente, ce qui n'est pas surprenant, les alarmes ajoutées n'étant spécifiques à aucun des défauts.

ii. Résultat Sim_{NWnorm}

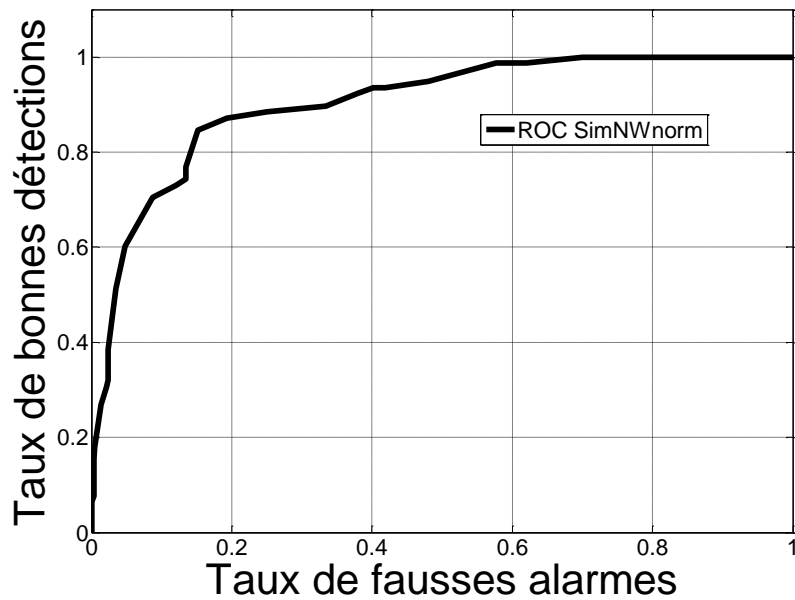


Figure 8-11 Représentation des prototypes séquentiels

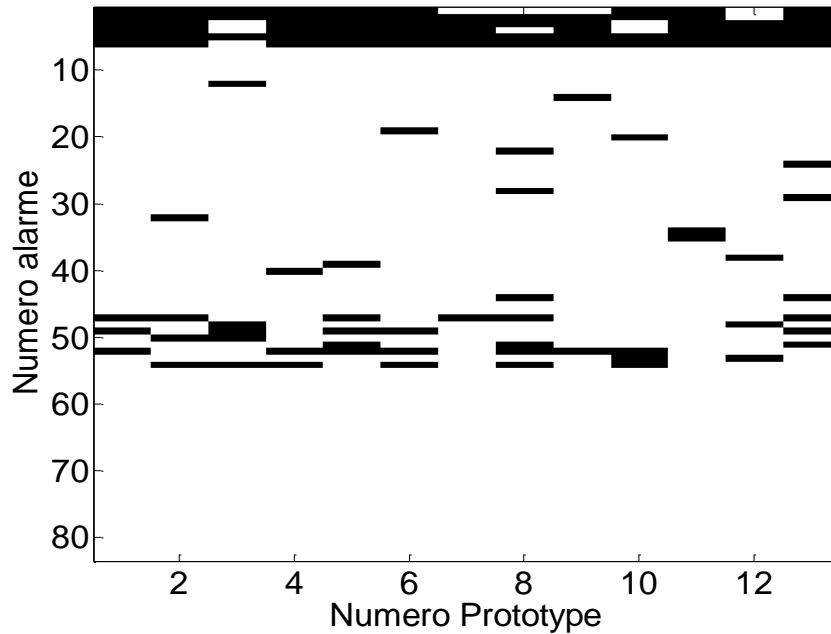


Figure 8-12 Courbe COR SimNwnorm

Observons maintenant les résultats avec Sim_{Nwnorm} sur la même base de données. On remarque qu'ici, les résultats sont toujours meilleurs que ceux utilisant D_{vp} . Cependant, la performance est fortement dégradée par rapport à la base précédente. La présence de cette séquence répétée qui apparaît dans tous les défauts a, en fait, un double effet sur les résultats du diagnostic.

Tout d'abord, la présence de ces séquences répétées va gonfler artificiellement la similarité des séquences entre elles de fait de leur présence dans tous les défauts de la base. Cet ajout de similarité va artificiellement créer des ressemblances entre les défauts, perturbant ainsi le bon diagnostic des défauts.

Ces séquences répétées perturbent également l'extraction des séquences constitutives dans les prototypes. On peut voir sur le graphique représentant les prototypes séquentiels extraits que les alarmes dont l'ordre est représentatif du défaut ne sont pas toujours extraites par l'algorithme. Ces défauts sont les défauts 3, 7, 8, 9, 10, 12. En effet, les alarmes répétées qui peuvent s'insérer entre les alarmes constitutives perturbent l'alignement et donc l'extraction de prototypes. Les schémas ci-dessous illustrent la façon dont une alarme appartenant à la séquence répétée peut perturber l'extraction d'un prototype. Considérons pour cela l'exemple de deux séquences appartenant au même défaut composé chacune d'une séquence constitutive et d'une séquence répétée. La séquence aléatoire est quant à elle volontairement non représentée pour des questions de simplification :

Soient les deux séquences appartenant au même défaut composé d'une séquence constitutive et d'une séquence répétée.

τ_1	γ_1	γ_2	τ_2	γ_3	τ_3
τ_1	τ_2	γ_1	γ_2	τ_3	γ_3

Un alignement optimal possible pour ces deux séquences est le suivant :

τ_1	X	γ_1	γ_2	τ_2	γ_3	τ_3	X
τ_1	τ_2	γ_1	γ_2	X	X	τ_3	γ_3

τ_1	X	γ_1	γ_2	X	X	τ_3	X
----------	---	------------	------------	---	---	----------	---

Figure 8-13 Illustration du mécanisme perturbant l'extraction des prototypes séquentiels

L'exemple ci-dessus montre comment la présence de séquences répétées peut empêcher l'extraction correcte de prototypes séquentiels contenant les alarmes intéressantes pour le diagnostic, c'est-à-dire celles formant la séquence constitutive représentée par les alarmes τ_z . La position dans lesquelles les alarmes de la séquence répétée sont présentes empêche l'alignement efficace des alarmes constitutives. En effet, dans l'exemple 1, les alarmes répétées γ_1 et γ_2 apparaissent entre τ_1 et τ_2 alors qu'elles apparaissent entre τ_2 et τ_3 dans l'exemple 2. L'alignement de séquence va donc aligner en priorité les alarmes γ_1 et γ_2 et « perdre » l'alignement de τ_2 . C'est ce phénomène qui empêche l'extraction correcte des séquences constitutives intéressantes pour le diagnostic.

Cependant, si ces alarmes sont connues comme étant déclenchées systématiquement par le système de contrôle, une simple opération sur la matrice des coûts permet d'augmenter considérablement les performances de diagnostic. Prenons pour matrice de coûts la matrice suivante :

$$C = \begin{pmatrix} 1 & -1 & \dots & \dots & \dots & -1 \\ -1 & \ddots & & & & \vdots \\ \vdots & & 1 & & & \vdots \\ \vdots & & & -1 & & \vdots \\ \vdots & & & & \ddots & \vdots \\ -1 & \dots & \dots & \dots & \dots & -1 \end{pmatrix} \begin{matrix} 1 \\ \vdots \\ 6 \\ \vdots \\ \vdots \\ 55 \end{matrix}$$

Les alarmes constitutives (numérotés de 1 à 6) ont une valeur de similarité de 1 alors que les alarmes répétées ont une valeur de similarité de -1. Le coût de substitution d'une alarme répétée avec elle-même est identique à celui de la substitution avec une autre

alarme. La courbe COR utilisant la nouvelle matrice de coût montre les nouveaux résultats sur la même base de données. Ceux-ci deviennent très bons. Ceci s'explique par le fait qu'en construisant la matrice de coûts de la sorte, on indique à l'algorithme qu'on ne désire pas aligner les alarmes autres que celles dont l'ordre est représentatif du défaut. On sanctionne ainsi les alarmes répétées, que l'on sait être non pertinentes. Les alarmes répétées ne sont ainsi pas extraites lors de la phase d'extraction de prototypes. La courbe COR tracée avec la nouvelle matrice de similarité ainsi que la représentation des prototypes séquentiels est exposée à la ci-dessous.

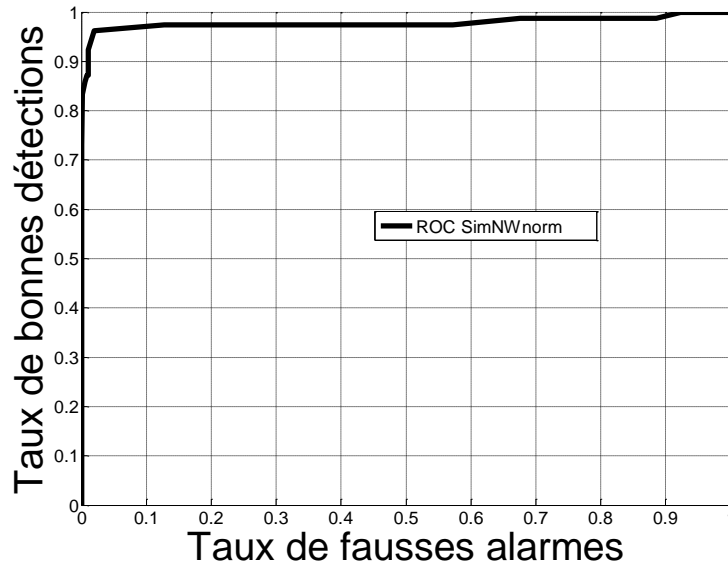


Figure 8-14 COR SimNwnorm base artificielle 2

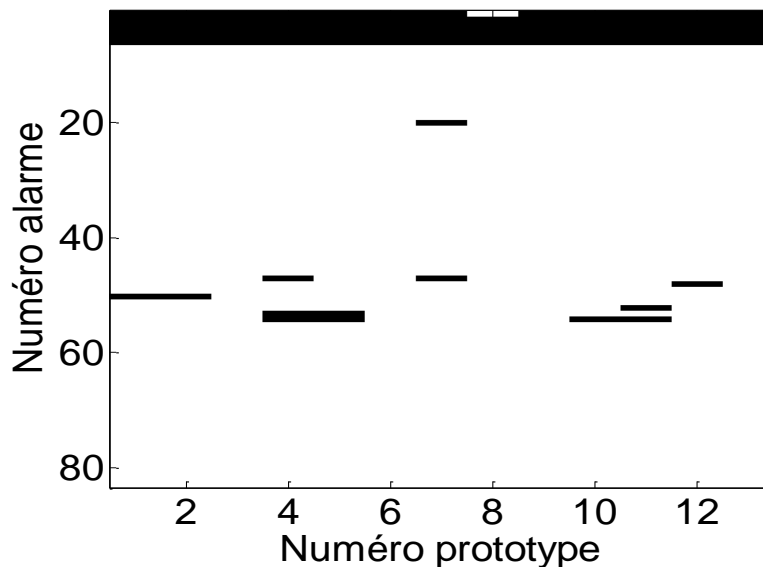


Figure 8-15 prototypes séquentiels base artificielle 2

L'observation des prototypes séquentiels montre que les alarmes répétées sont pour la grande majorité absente du prototype séquentiel, ce qui a deux effets aidant au diagnostic : Les prototypes séquentiels extraits sont meilleurs : les alarmes discriminantes numérotés de 1 à 6 sont bien plus présentes dans les prototypes séquentiels. Il manque au maximum 1 alarme parmi les 6. L'autre effet positif réside au niveau du calcul de la distance car pénaliser ces alarmes par une valeur de -1 permet d'éviter d'augmenter artificiellement la similarité.

8.1.2.3. Base contenant des séquences constitutives, des séquences aléatoires et des séquences uniques

Étudions les résultats maintenant dans la base définie avec les paramètres suivants : $f = 13, k = 6, T = 6, \theta = 40, \Gamma = 0$ et $\Psi = 2$. Cette base correspond à une base contenant une séquence constitutive pour chaque défaut, une séquence aléatoire, une séquence répétée et une séquence unique. Une représentation sous forme vectorielle de cette base est présentée ci-dessous :

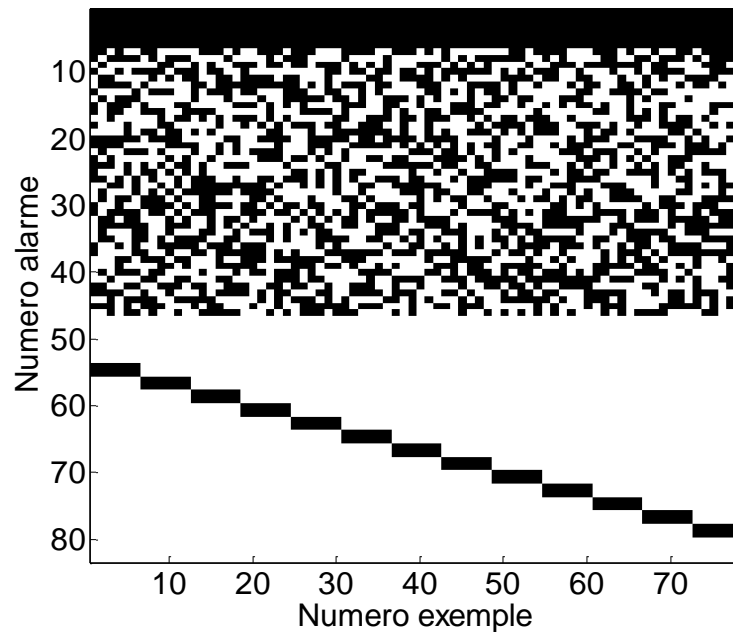


Figure 8-16 Représentation base artificielle 3

Les courbes COR pour les deux méthodes sont présentées dans la figure ci-dessous.

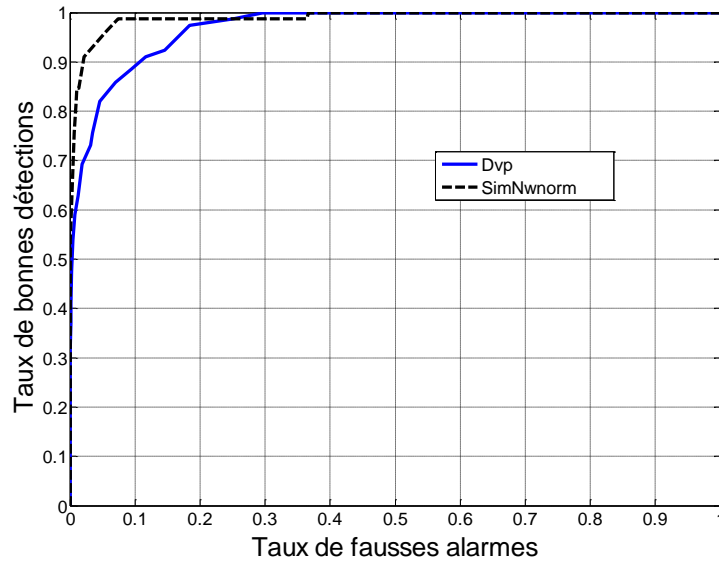


Figure 8-17 Courbes COR D_{vp} et Sim_{Nwnorm}

Les deux classifieurs obtiennent de bonnes performances. On se trouve en effet dans une configuration favorable aux deux algorithmes. Il existe des alarmes uniques qui permettent à D_{vp} de réaliser de bons résultats et également une séquence constitutive avec un ordre représentatif pour chaque défaut qui permet de bonnes performances pour Sim_{Nwnorm} . On note toutefois de meilleures performances pour Sim_{Nwnorm} qui semble mieux se comporter dans cette configuration.

8.1.2.4. Base contenant des séquences constitutives, des séquences aléatoires, des séquences répétées et des séquences uniques

Étudions les résultats maintenant dans la base définie avec les paramètres suivants : $m = 13, k = 6, T = 6, \Theta = 40, \Gamma = 8$ et $\Psi = 2$. Ceci correspond à une base dans laquelle tous les défauts sont constitués de chacun des quatre types de séquences définies au paragraphe 2. La Figure 8-18 ci-dessous donne la représentation vectorielle de la base complète.

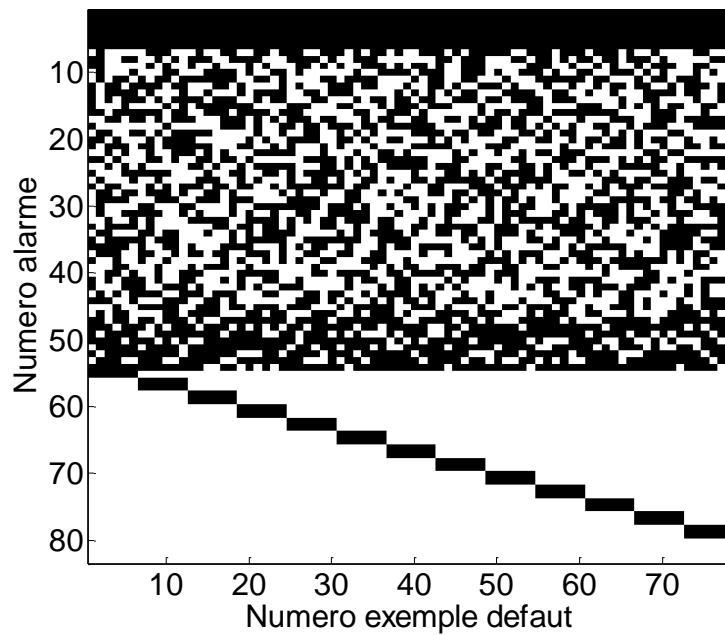


Figure 8-18 Base artificielle complète

Les résultats sur cette base sont présentés dans la Figure 8-19 ci-dessous, à l'aide de la courbe COR.

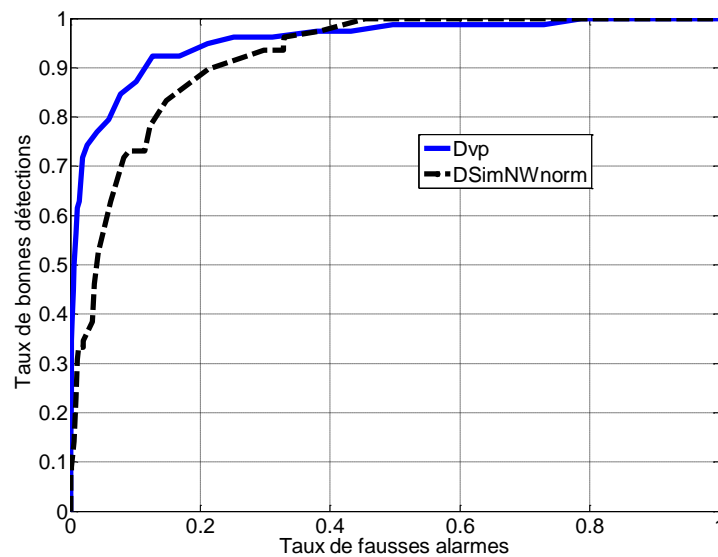


Figure 8-19 COR Dvp et SimNWnorm

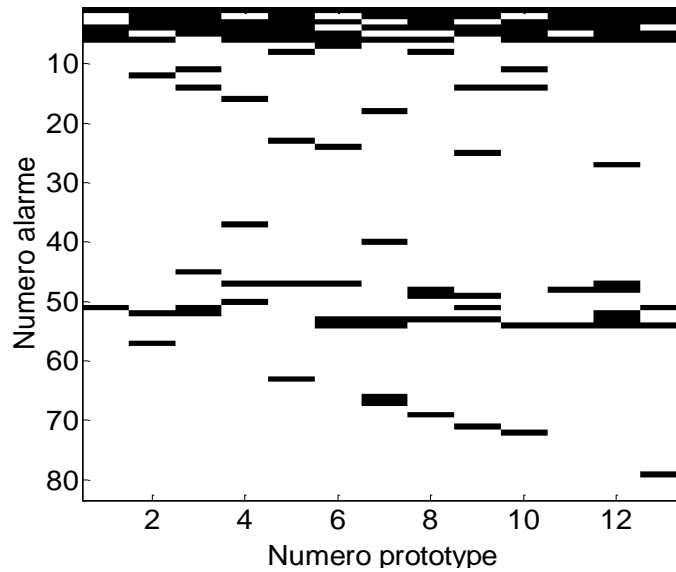


Figure 8-20 Prototypes séquentiels base complète

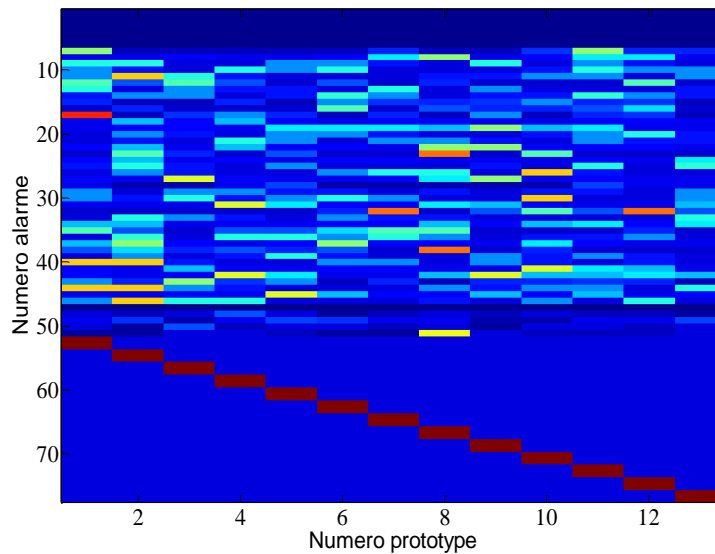


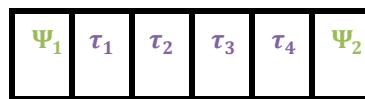
Figure 8-21 Poids Dvp

L'analyse des courbes COR montre que la distance D_{vp} se comporte légèrement mieux que Sim_{NWnorm} . L'analyse des prototypes séquentiels et des poids de la méthode de D_{vp} peut expliquer cette différence de résultats. La méthode vectorielle permet d'extraire avec succès les alarmes caractéristiques ψ (au nombre de deux dans notre exemple) de chaque défaut (alarmes numérotés de 54 à 80) à l'aide des poids de chacun d'eux. L'algorithme d'extraction de prototype séquentiel a plus de difficultés dans la tâche d'extraction de ces alarmes. Un seul prototype séquentiel contient les deux alarmes uniques. Les autres prototypes ne contiennent qu'une, voire 0 alarme unique. Ceci s'explique par le fait que ces alarmes ne sont

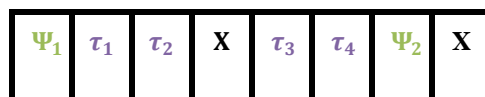
pas toujours placées dans les mêmes positions selon les exemples du défaut, elles peuvent « s'intercaler » entre deux alarmes constitutives (alarmes numérotés entre 1 et 6) et ne plus apparaître dans le prototype séquentiel. Un schéma du mécanisme est présenté ci-dessous :

Soient deux exemples du même défaut constitués d'une séquence constitutive et d'une séquence unique, pour des raisons de simplification les séquences aléatoires et répétées ne sont pas représentées:

Soient deux exemples du même défaut :



Soit un alignement optimal possible pour ces deux séquences :



Le prototype extrait de cet alignement est le suivant :



Figure 8-22 Mécanisme d'extraction des alarmes uniques

Les alarmes uniques ne sont pas toutes présentes. Par exemple l'alarme ψ_1 est absente du prototype séquentiel. L'alarme constitutive τ_4 est aussi absente du prototype alors qu'elle est également utile au diagnostic. Ainsi, les alarmes toujours présentes dans les séquences d'un même défaut mais dont l'ordre d'apparition dans la séquence n'est pas toujours le même (alarmes uniques ou alarmes répétées) perturbent le mécanisme d'extraction de prototypes séquentiels. A l'inverse, la méthode vectorielle filtre correctement ces alarmes répétées grâce au calcul des poids. Notons cependant que l'utilisation d'une matrice de coût adaptée permettrait de résoudre ce problème comme montré dans le paragraphe 8.1.2.2.

8.1.3. Conclusions

L'analyse des différentes bases permet de mettre en relief les différentes configurations et distinguer celles favorables à l'une ou à l'autre des méthodes vectorielle ou séquentielle. Si les listes d'alarmes ne présentent pas d'alarmes réellement caractéristiques (spécifiques à un défaut et pas à un autre), la méthode vectorielle n'obtient pas de bons résultats. La méthode séquentielle permet, elle, de discriminer les défauts dans la mesure où l'ordre d'apparition des alarmes communes à tous les défauts varie d'un défaut à l'autre. Cependant, elle est sensible à la présence d'alarmes communes à tous les défauts dont l'ordre d'apparition se répète systématiquement. Cette difficulté peut cependant être surmontée à l'aide d'une matrice de coût appropriée qui permet de sanctionner ce type d'alarmes avec un coût négatif. D_{vp} se montre plus efficace pour éliminer ces alarmes à l'aide du calcul de poids mais elle ne peut fonctionner que s'il existe des alarmes distinctes pour chaque défaut. Ces alarmes peuvent également être un atout pour Sim_{NWnorm} à condition que leurs positions relatives par rapport aux alarmes constitutives ne varient pas.

8.2. Études sur la base de données du CERN

On rappelle que l'on dispose de $m=13$ défauts différents, avec $k=6$ exemples par défaut. Les défauts ont déclenché $N=97$ alarmes différentes, ce qui nous donne donc un alphabet Ω de taille $2 * N = 194$.

8.2.1. Influence des paramètres ω et λ

Cette analyse est réalisée à partir des courbes COR, tracées en affectant différentes valeurs au paramètre λ puis ω . Chaque courbe COR est obtenue pour une valeur de ω et de λ , en faisant varier le seuil de détection T_d . λ peut varier entre $3/5$ et $5/5$, étant donné que l'on dispose de 6 exemples pour chaque défaut et que l'on crée des prototypes à l'aide de 5 exemples. La valeur de ω varie entre 0 et -1 compte tenu de la matrice des coûts choisie. Rappelons que celle-ci vaut -1 sur tous les éléments de la matrice à l'exception de la diagonale où les éléments valent 1. La valeur de 0 signifie que le coût d'ouverture d'un gap est nul et la valeur de -1 qu'elle est égale à celle du coût de substitution de deux alarmes.

8.2.2. Courbes COR en fonction de λ

On trace dans la Figure 8-23 ci-dessous les courbes COR obtenues en faisant varier λ et en fixant la valeur de $\omega = 0$.

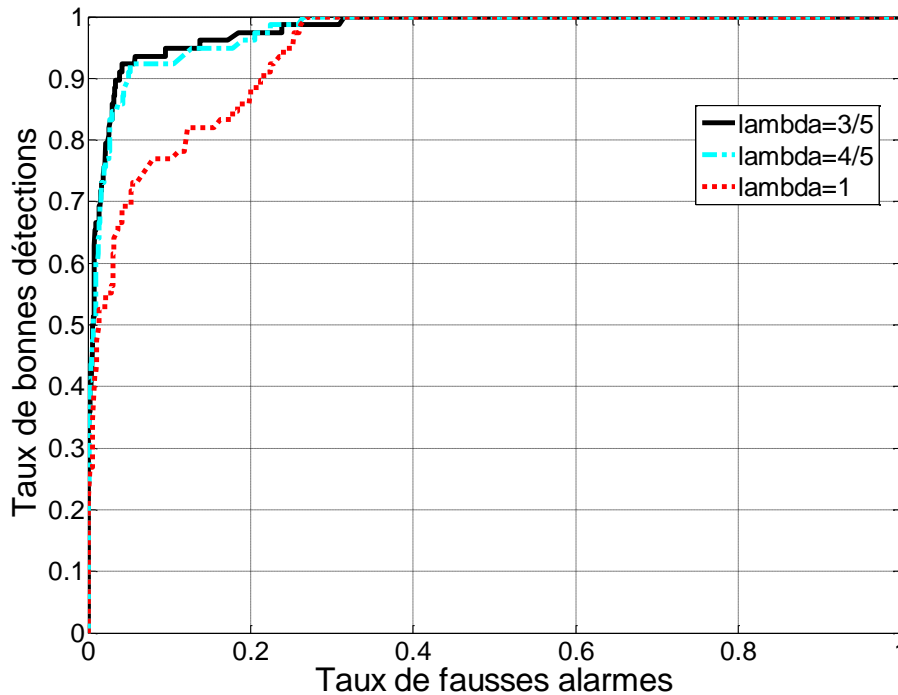


Figure 8-23 Courbe COR en fonction lambda

L'analyse des courbes montre que les résultats sont similaires pour $\lambda = 3/5$ et $\lambda = 4/5$. On note des performances très dégradées pour une valeur de $\lambda = 1$. Ceci est dû à un filtrage trop important des alarmes et une perte d'information essentielle au bon diagnostic. Les longueurs moyennes des séquences extraites montrent cette perte d'informations. Avec $\lambda = 3/5$, la longueur moyenne des prototypes est de 38 alarmes, ce qui est à peu près identique à la longueur moyenne des séquences extraites avec $\lambda = 4/5$, qui est de 35. Avec $\lambda = 1$, la longueur des prototypes passe à 25 soit une différence d'environ 30% sur le nombre d'alarmes constituant la séquence. Notons les bonnes performances obtenues avec $\lambda = 3/5$ et 4, avec un taux bonnes détections de 93% pour un taux de fausses alarmes de l'ordre de 5%.

8.2.3. Courbes COR en fonction de ω

De la même façon, on étudie l'évolution de la courbe COR en fonction de la valeur de ω , celle-ci variant entre 0 et -1. La Figure 8-24 montre le tracé de ces courbes en fonction de ω avec une valeur de $\lambda = 3/5$.

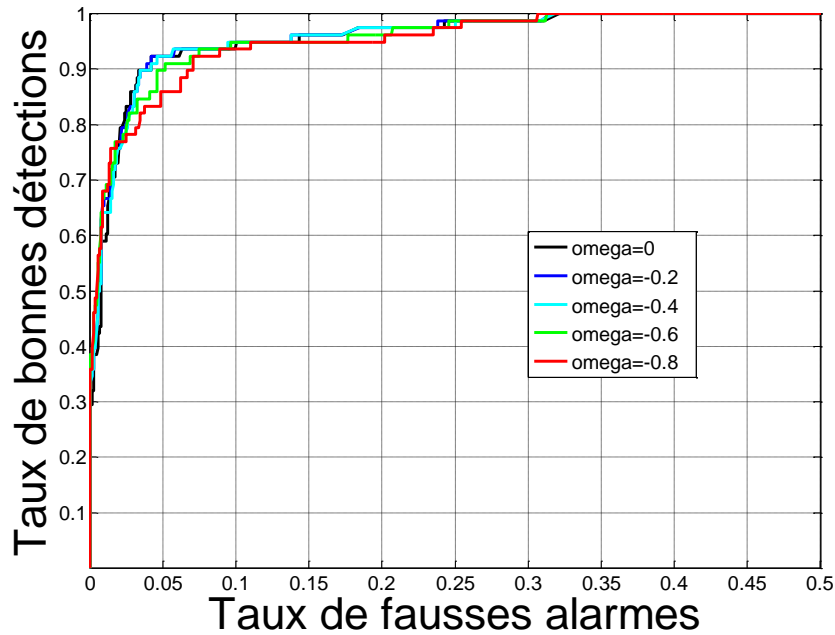


Figure 8-24 Courbe COR en fonction de ω

Ici, les meilleurs résultats sont obtenus pour des valeurs de pénalités proches de 0 (les courbes COR pour ω égale 0, -0.2 et -0.4 se chevauchent). Ceci témoigne d'une base relativement bruitée (contenant un nombre important d'alarmes non pertinentes) nécessitant de nombreuses manipulations pour extraire des prototypes représentatifs du défaut, ce qui justifie l'intérêt de peu pénaliser l'introduction d'un gap. On note également une dégradation des performances entre les pénalités allant de -0.6 jusqu'à -1. Cependant, ces courbes COR montrent que la méthode est relativement peu sensible au paramètre ω .

L'analyse des différents résultats met en valeur les effets des paramètres de réglages que sont λ et ω . L'analyse montre des paramètres relativement peu sensibles aux variations des différents paramètres à l'exception des valeurs extrêmes, ce qui minimise le risque d'erreur lors du paramétrage et assure une faible dépendance à la base d'apprentissage.

8.2.4. Comparaison avec la méthode vectorielle, D_{vp}

8.2.4.1. Courbe COR

On trace comme précédemment la courbe COR pour évaluer les performances de la méthode lorsque la décision est prise par seuillage. Les tests sont réalisés avec le paramétrage suivant $\lambda = 3/5$ et $\omega = 0$ car il s'agit du des paramétrages donnant le meilleur résultat sur la courbe COR comme vu précédemment. À titre de comparaison, on trace pour rappel, la courbe COR de D_{vp} .

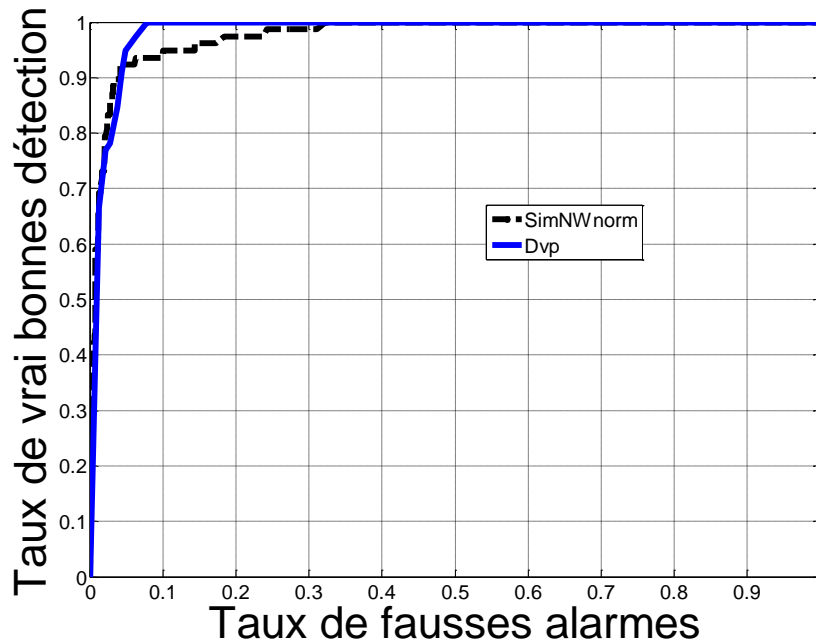


Figure 8-25 Courbe COR D_{vp} et Sim_{NWnorm} base simulateur

On peut observer les bons résultats observés avec Sim_{NWnorm} . En effet, avec un taux de fausses alarmes de l'ordre de 5% on atteint 93% de bonnes détections de l'ordre. Cependant D_{vp} atteint plus rapidement les 100% de taux de bonnes détections que Sim_{NWnorm} . En effet, ce taux est atteint pour une valeur de 10% de taux de fausses alarmes pour D_{vp} alors qu'il faut accepter 35% de fausses alarmes pour atteindre un taux de 100% de bonnes détections. Cette différence réside dans la difficulté de la méthode séquentielle à diagnostiquer 6 exemples de défauts sur les 78 de la base. On étudiera plus loin la différence de performances des deux méthodes en détaillant sur les défauts correctement diagnostiqués et ceux mal diagnostiqués par l'algorithme.

8.2.4.2. Matrices de Confusion

On trace ici la matrice de confusion de Sim_{NWnorm} et D_{vp} pour un seuil de détection T_d configuré à 90% de bonnes détections, ce qui correspond au seuil où les deux méthodes présentent des résultats similaires en termes de taux de fausses alarmes.

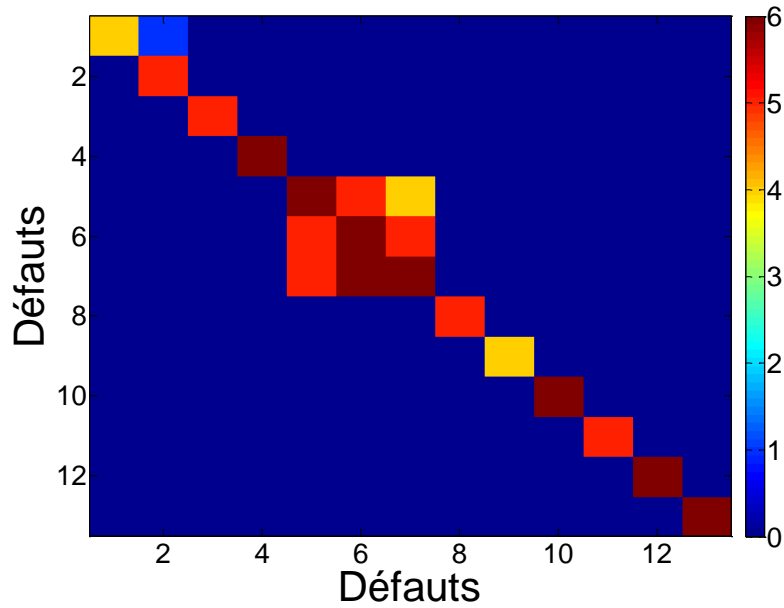


Figure 8-26 Matrice de confusion $SimNWnorm$ $T_d = 90\%$

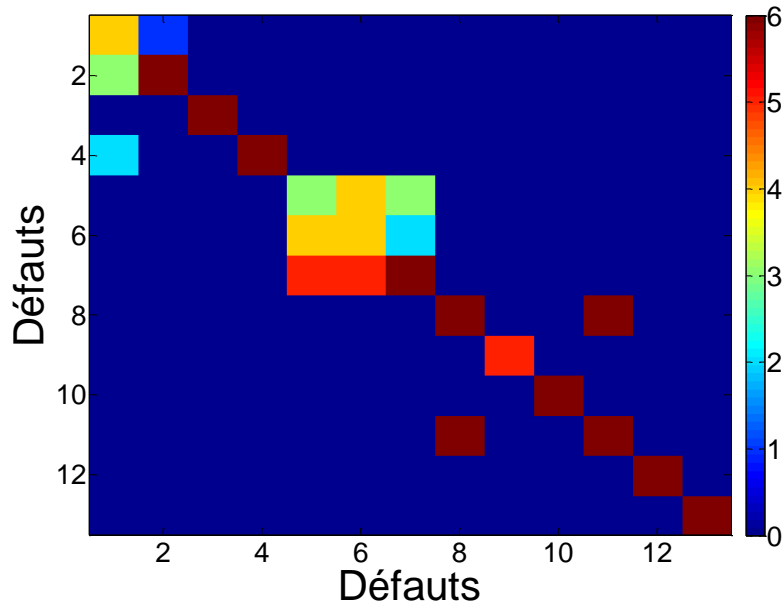


Figure 8-27 Matrice de confusion Dvp $T_d = 90\%$

On remarque que les deux algorithmes se comportent de manière identique pour la plupart des défauts de la base, notamment pour les groupes de défauts qui sont confondus ((1, 2, 3) et (5, 6, 7)). Cependant il existe deux défauts systématiquement confondus par la Dvp et jamais par Sim_{NWnorm} . Il s'agit du groupe du groupe de défauts numéroté 8 et 11. On étudiera ces différents cas dans le paragraphe suivant. On observe figure 8-29 ci-dessous le nombre de propositions moyen fait par chacune des distances, la courbe bleue correspond à Dvp et la courbe noire à Sim_{NWnorm} :

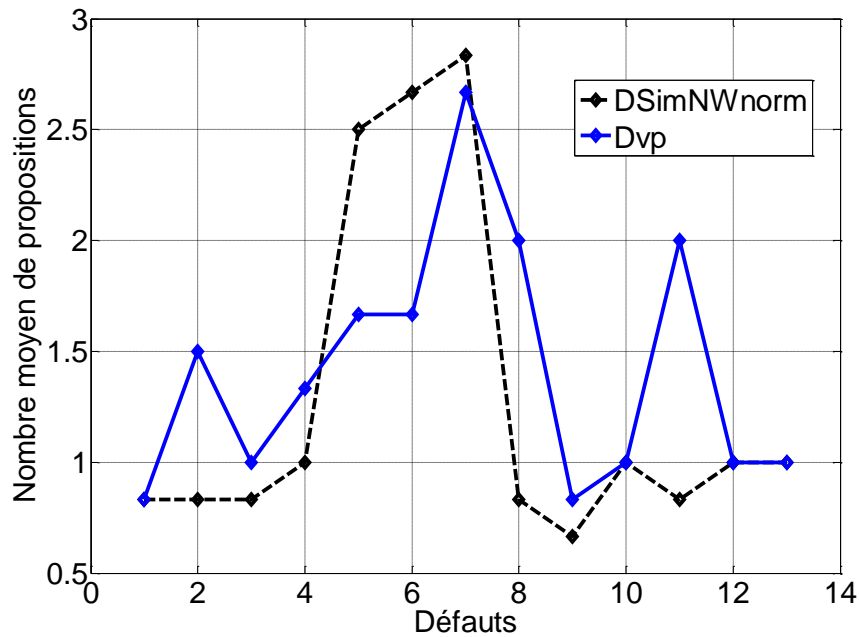


Figure 8-28 Nombre de propositions moyens en fonction du défaut $T_d=90\%$

On voit ici que les principales différences de performances se situent pour le groupe de défaut (1,2,3,4) qui est fortement confondu lorsque D_{vp} est utilisé mais pas lorsque Sim_{NWnorm} est utilisée. La seconde différence réside dans la confusion des défauts 8 et 11 qui sont confondus par D_{vp} et non pas par Sim_{NWnorm} . Les résultats sont équivalents pour les autres groupes de défauts.

On analyse à présent les résultats sur la matrice de confusion avec un niveau de détection plus élevé en posant $T_d = 98\%$. On a vu que les résultats étaient alors meilleurs en utilisant D_{vp} . Observons cela sur la matrice de confusion :

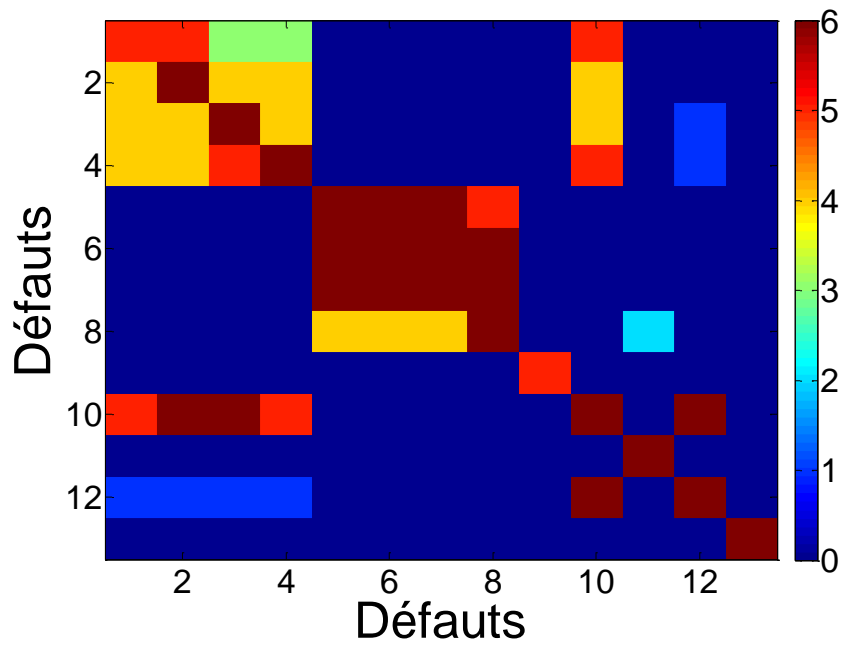


figure 8-29 $Sim_{NWnorm} T_d = 98\%$

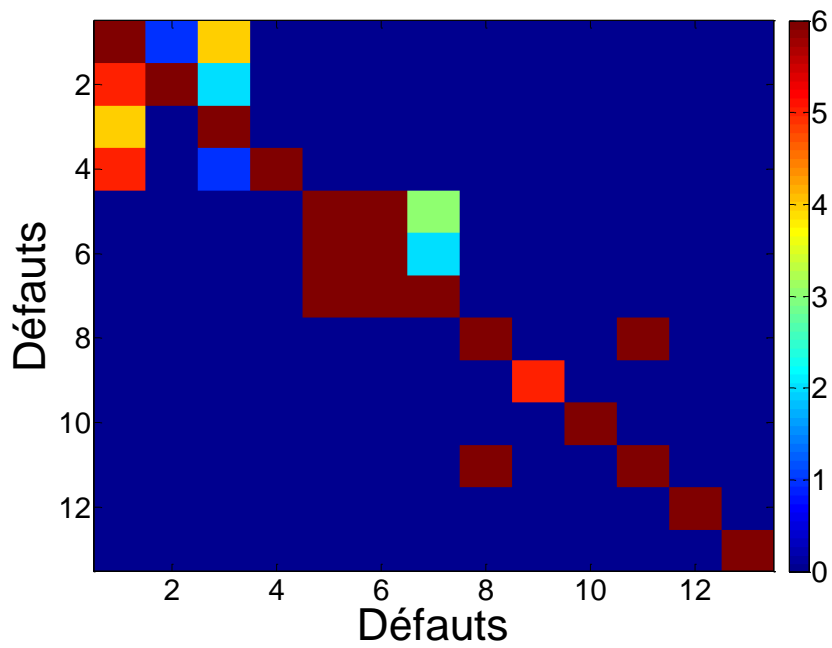


figure 8-30 $D_{vp} T_d = 98\%$

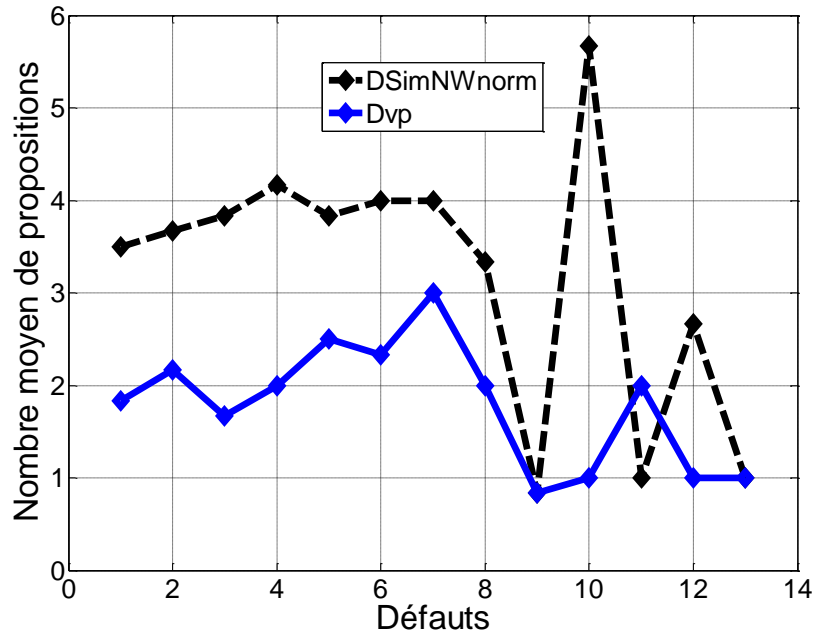


Figure 8-31 Nombre de propositions moyens en fonction du défaut $Td=98\%$

L'analyse des courbes COR montre une forte dégradation des performances lorsque le seuil de détection est fixé à 98%, ce qui est en accord avec la courbe ROC. On a vu qu'au-delà de 90% de bonnes détections, les performances des deux méthodes divergeaient. Une analyse des différents comportements des méthodes sera présentée plus loin afin de comprendre cette différence de résultats. On note toutefois que les défauts 8 et 11 sont confondus avec D_{vp} mais que Sim_{NWnorm} arrive à les différencier. Ce cas sera étudié dans la suite pour comprendre l'apport de Sim_{NWnorm} , alors que les cas des défauts 10 et 12 seront eux étudiés pour comprendre l'apport de D_{vp} par rapport à Sim_{NWnorm} . Notons cependant que Sim_{NWnorm} obtient des performances supérieures à la distance D_h et D_{ch} pour l'ensemble des défauts, excepté le défaut 10 où le nombre de propositions faites à l'opérateur reste égal à 6 (cf Figure 6-21).

8.2.4.3. Performances en rejet

L'étude des performances en rejet de la méthode s'étudie d'abord à l'aide de la courbe COR en rejet. La courbe est construite de manière identique à celle de D_{vp} (voir paragraphe 6.3.3.2). On présente le taux de bonnes détections. La

Figure 8-32 présente la courbe COR en rejet des distance D_{vp} et Sim_{NWnorm} .

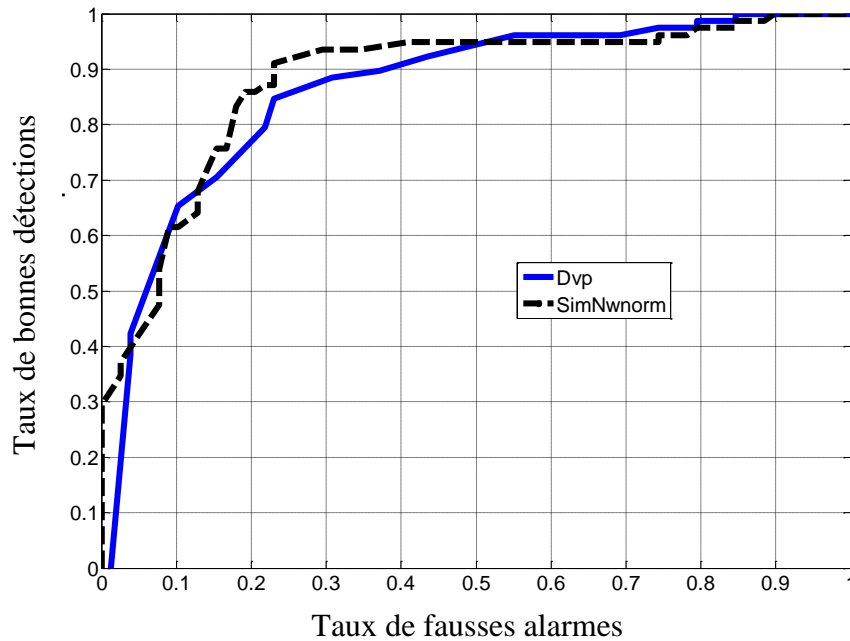


Figure 8-32 COR rejet Sim_{Nwnorm} et D_{vp}

L'analyse de ces deux courbes met en évidence une légère meilleure capacité de rejet de la méthode Sim_{Nwnorm} par rapport à D_{vp} . Pour un taux de bonnes détections de 90 %, le taux de fausses alarmes est limité à 23% alors qu'il est de l'ordre de 40% avec D_{vp} . Il faudra choisir un taux de bonnes détections élevé car de ce taux dépend le pourcentage maximum de bons diagnostics de la méthode. Si l'on choisit un taux de 90%, on aura un taux de bon diagnostic au maximum de 90% car 10% des défauts seront déjà exclus avec le seuil du rejet T_r alors que leur prototype est bien contenu dans la base. On rappelle que, lors du rejet, seuls les exemples d'un même défaut sont pris en compte, soit lors du calcul des poids pour D_{vp} (les poids sont calculés selon $w_j^i = (2 * \alpha_j^i - 1)$), mais également pour Sim_{Nwnorm} , de par sa définition. Or, nous avons vu dans le chapitre 6 que l'apport de la méthode D_{vp} réside en grande partie dans la prise en compte des exemples des autres défauts lors du calcul des poids, d'où des résultats en rejet en deçà de ce qui peut être attendu. Ainsi Sim_{Nwnorm} , en prenant en compte l'ordre d'apparition des alarmes, réalise de meilleurs résultats lors du rejet alors que ses résultats sont inférieurs à ceux de D_{vp} en localisation.

8.3. Comparaison des méthodes D_{vp} et Sim_{Nwnorm}

Dans ce paragraphe, nous comparons les deux mesures de D_{vp} et Sim_{Nwnorm} , en évaluant dans quelle situation chaque méthode est la plus adaptée. On se servira pour cela des analyses faites sur les matrices de confusions, en se basant sur les défauts qui sont confondus par D_{vp} et pas par Sim_{Nwnorm} et inversement.

8.3.1. Apport de l'ordre d'apparition des alarmes dans l'aide au diagnostic

Étudions les cas des défauts numérotés 8 et 11 qui sont confondus à de nombreuses reprises lors de l'analyse avec la méthode vectorielle D_{vp} et qui ne le sont jamais en utilisant Sim_{NWnorm} . On propose d'observer les prototypes séquentiels et vectoriels de ces deux défauts et de les comparer. La Figure 8-33 représente les prototypes vectoriels des défauts 8 et 11 mis côte à côte en utilisant la représentation graphique sous forme d'image utilisée auparavant dans ce chapitre.

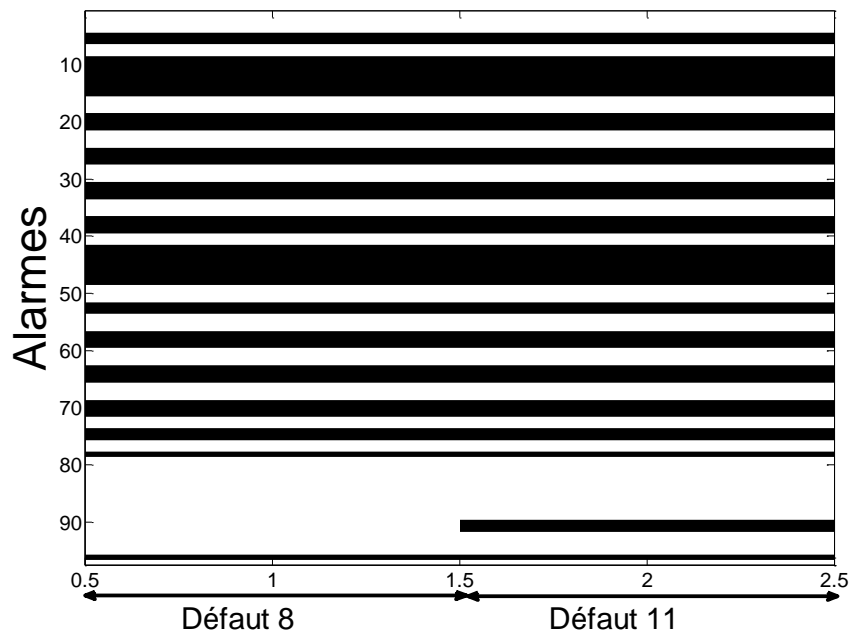


Figure 8-33 Prototypes vectoriels défauts 8 et 11

L'observation des deux prototypes met en relief de nombreuses alarmes communes aux deux prototypes vectoriels, il n'existe que deux alarmes propres aux défauts 11 qui ne sont pas suffisantes pour les différencier. Observons maintenant les prototypes séquentiels des défauts 8 et 11 extraits à l'aide de l'algorithme séquentiel. Les deux prototypes séquentiels seront représentés sous forme graphique pour une meilleure compréhension de la séquence. Le graphique sera construit de la manière qui suit : les alarmes déclenchées seront affichées sur un axe de représentation en deux dimensions. L'ordonnée correspond au numéro de l'alarme et l'abscisse correspond à la position de l'alarme dans la séquence. L'espace des alarmes possibles est réduit aux alarmes apparaissant dans les prototypes séquentiels 8 et 11.

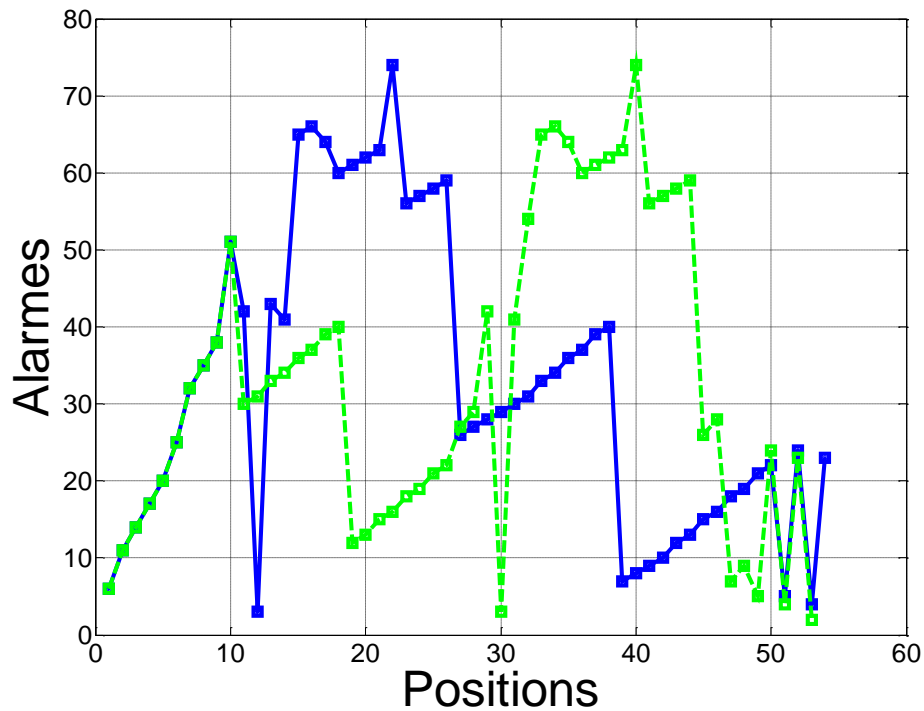


Figure 8-34 Représentation graphique des prototypes séquentiels 8 et 11

La Figure 8-34 montre les prototypes séquentiels des défauts 8 et 11 sur le même graphique avec, en bleu, les alarmes appartenant au défaut 8 et, en vert, les alarmes du défaut 11.

On décompose la Figure 8-34 afin d'en extraire les différents groupes d'alarmes et d'observer les prototypes séquentiels et l'ordre d'apparition des groupes séquentiels. Chaque groupe d'alarmes est désigné par une lettre de l'alphabet. Les groupes appartenant au défaut 8 sont montrés en lignes pleines et ceux du défaut 11 en ligne pointillées (excepté pour le groupe nommé A qui est commun aux deux défauts).

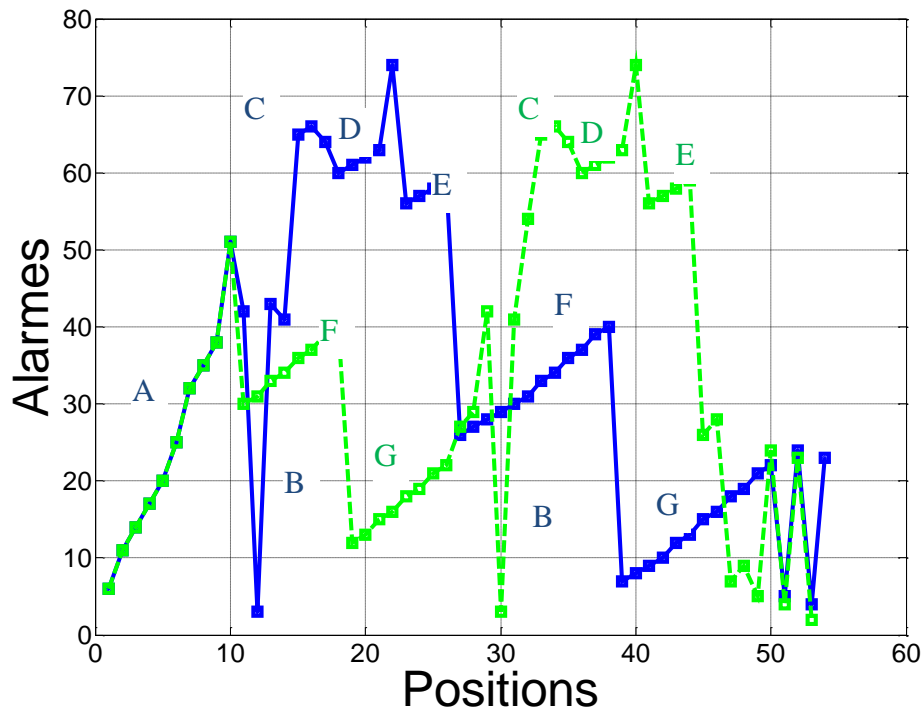


Figure 8-35 Représentation graphique des prototypes séquentiels 8 et 11 avec appellation des groupes d'alarmes

On peut observer que le prototype séquentiel du défaut 8 peut s'écrire de manière assez grossière si on excepte les alarmes isolés $P^8 = \langle A, B, C, D, E, F, G \rangle$. Celui du défaut 11 peut s'écrire de la manière qui suit $P^{11} = \langle A, F, G, B, C, D, E \rangle$. Ici, les mêmes alarmes sont présentes dans les deux prototypes mais leur ordre est complètement différent, mis à part le premier et le dernier groupe d'alarmes. C'est cette différence dans l'ordre d'apparition, qui ne peut se détecter qu'à l'aide de prototypes séquentiels qui permet la discrimination de ces deux défauts. La majorité des alarmes sont des alarmes que nous avons appelées « constitutives » précédemment.

Les défauts 8 et 11 correspondent à des types de pannes similaires. Il s'agit, dans les deux cas, d'une panne bloquant la circulation correcte du mélange de gaz et entraînant une hausse de la pression à l'intérieur du circuit principal. Le défaut 8 correspond à un régulateur de débit fermé qui empêche le gaz de s'échapper, ce qui fait ainsi augmenter la pression dans tout le système. Le défaut 11 correspond à une panne sur le module pompe responsable de la circulation du gaz dans le système. Une panne dans ce module augmente également la pression dans tout le système, d'où des listes d'alarmes très similaires pour ces deux défauts. Cependant, ces deux défauts ont lieu dans des modules très éloignés l'un de l'autre dans le circuit (l'un dans le module « Echappement » l'autre dans le module « Pompe »), ce qui engendre une propagation différente du défaut et donc un déclenchement dans un ordre différent des alarmes. C'est dans ce type de situation là qu'utiliser Sim_{NWnorm} est un atout dans le diagnostic. L'état final du système est identique mais le chemin pour atteindre cet état

est différent suivant le défaut, il faut alors privilégier une représentation séquentielle des défauts.

8.3.2. Apport de D_{vp} par rapport à Sim_{NWnorm}

Étudions le cas des défauts (10,12) qui sont confondus par Sim_{NWnorm} et pas par D_{vp} . On propose une analyse des prototypes séquentiels afin de mieux comprendre pourquoi les deux défauts sont confondus lors de l'analyse par Sim_{NWnorm} . Le prototype du défaut numéro 10 est lui représenté en bleu et le prototype du défaut numéro 12 en vert.

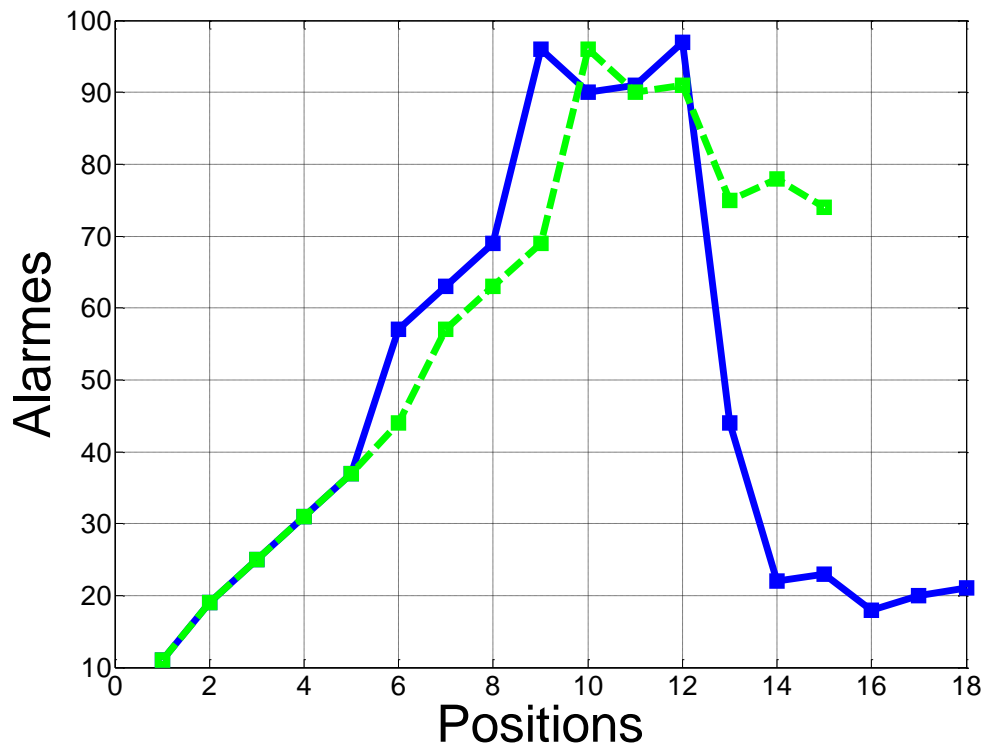
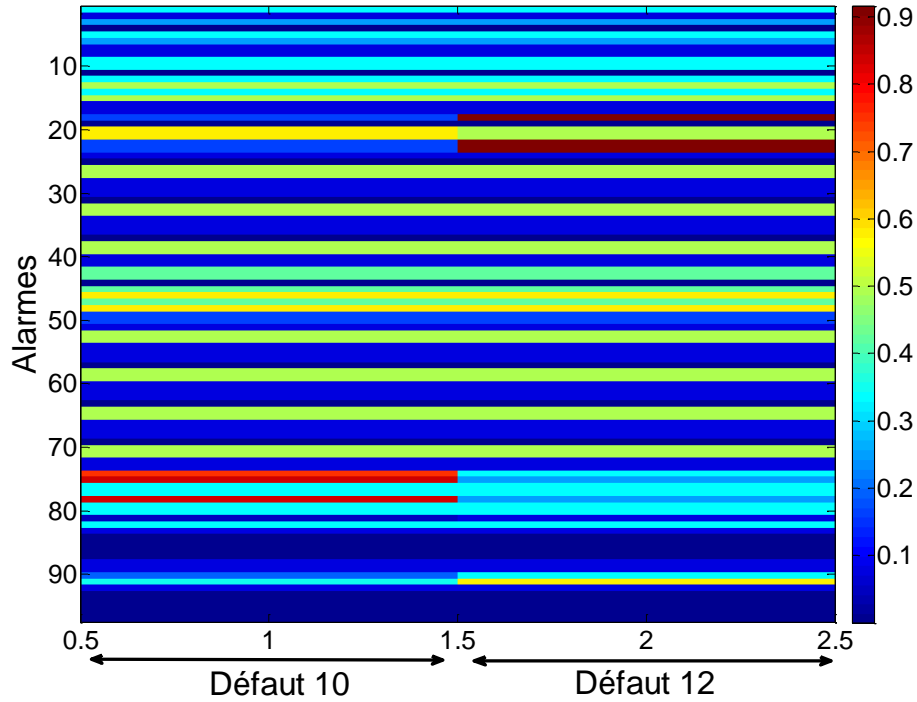


Figure 8-36 Prototypes séquentiels des défauts 10 et 12

L'étude des prototypes séquentiels montrent une forte similarité entre les deux. En effet, les 11 premières alarmes sont identiques et suivent toujours la même séquence pour les deux prototypes. Ils existent cependant certaines alarmes spécifiques à chacun des défauts mais leur influence respective dans le calcul de la similarité est identique aux 11 premières alarmes. Celles-ci augmentent la similarité de façon artificielle, menant à la confusion des deux défauts dans la plupart des cas.

Étudions maintenant les poids des alarmes calculés avec D_{vp} .



*Figure 8-37 Poids $w_j^i = (2 * \alpha_j^i - 1) * (1 - \beta_j^i)$ défaut 10 et 12*

On voit sur la matrice des poids $w_j^i = (2 * \alpha_j^i - 1) * (1 - \beta_j^i)$ que les alarmes qui sont propres aux défauts 10 et 12 sont mises en valeur avec des poids élevés de l'ordre de 0.9 alors que celle des alarmes qui sont partagées sont de l'ordre de 0.5 voire moins. Cette différence permet à D_{vp} de différencier ces deux défauts en augmentant de façon significative la distance entre les deux eux. On notera que les alarmes mises en relief par les poids sont celles que l'on retrouve dans les prototypes séquentiels et qui sont propres à chaque défaut. Elles sont numérotées (74,75,78) pour le défaut 10 et (18,22,23) pour le défaut 12.

8.3.3. Étude des erreurs de diagnostics

A l'instar de ce qui a été fait dans le paragraphe précédent, on propose d'étudier les prototypes séquentiels des défauts 5 et 6 qui sont confondus quelle que soit la méthode (D_{vp} , D_{ch} et D_h ou Sim_{NWnorm}). La Figure 8-38 présente les prototypes séquentiels des deux défauts superposés sur un même graphe en utilisant la même représentation que précédemment. Les alarmes du prototype 5 sont représentées en bleu et ceux du prototype 6 en vert.

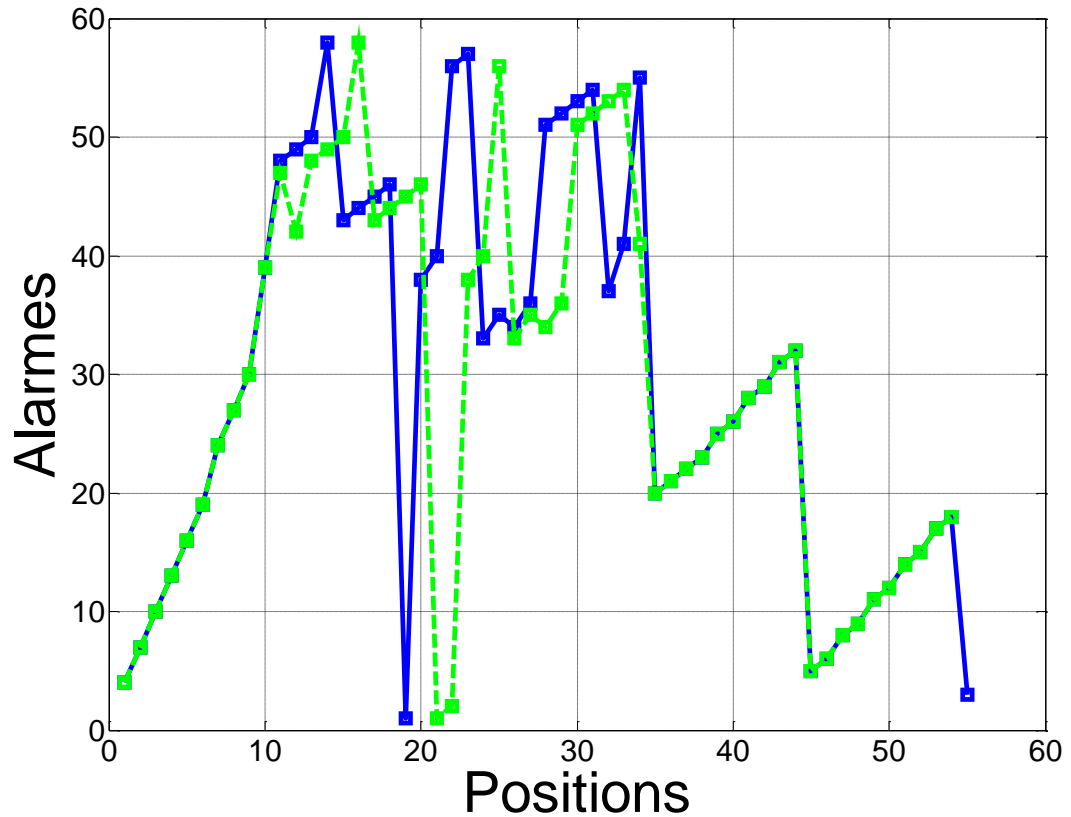


Figure 8-38 Prototypes séquentiels défaut 5 et 6

Ici, les prototypes des deux défauts sont fortement similaires. On retrouve les mêmes alarmes en majorité déclenchées pour les deux défauts (ce qui a pu déjà s'observer lors de l'analyse vectorielle) mais on remarque de plus que l'ordre de déclenchement de ces alarmes est aussi identique pour les deux prototypes, à l'exception de quelques alarmes isolées dont l'ordre diffère entre les deux prototypes. Ces alarmes ne sont pas assez nombreuses comparées à toutes les autres alarmes qui gardent le même ordre d'apparition dans les deux prototypes pour discriminer les deux défauts. Ce cas correspond au cas où le nombre d'alarmes toujours dans le même ordre dans les défauts est grand (séquences répétées, alarmes notées γ_z avec un Γ grand, dans la base artificielle) et met en défaut le diagnostic. Les deux défauts ne sont pas dissociables que ce soit avec la méthode vectorielle (pas ou peu d'alarmes uniques, Ψ faible voir nul), ni par la méthode séquentielle (beaucoup de séquences répétées, Γ grand).

Les défauts 5 et 6 sont des défauts similaires qui correspondent tous deux à une fuite de gaz, l'une dans le module « Mixeur » (défaut 5) et l'autre dans le module « Echappement » (défaut 6). Même si ces défauts ont lieu dans des modules distincts, ils sont en réalité très proches physiquement si l'on suit le parcours du gaz à travers le circuit. De plus, il existe peu d'instrumentation entre les deux points de défauts qui permettrait d'observer ce qui peut se passer entre ces deux points. Le même type de défaut déclenchant les mêmes alarmes

combiné à une proximité physique provoquant un enchaînement d'alarmes identique font de ces défauts deux défauts difficiles à différencier par D_{vp} ou Sim_{NWnorm} .

8.3.4. Conclusion sur la base de données du CERN

L'analyse des résultats permet de mettre en relief les propriétés des défauts représentées dans la base du CERN. Ils se décomposent en 3 grandes familles. La première est composée des défauts ayant plusieurs alarmes spécifiques et dont l'ordre d'apparition est le même. Ces défauts sont simples à diagnostiquer, les méthodes vectorielle et séquentielle sont toutes les deux adaptées. Il existe également un autre groupe de défauts, représenté par les défauts 8 et 11 que nous avons analysés dans le paragraphe 8.3.3 qui possèdent un grand nombre d'alarmes similaires mais dont l'ordre d'apparition permet de les différencier et pour lesquels Sim_{NWnorm} est plus adaptée. D'autres types de défaut sont mieux diagnostiqués par D_{vp} , il s'agit de ceux dont l'ordre d'apparition des alarmes est toujours le même, et pour lesquels il existe peu d'alarmes spécifiques. Nous avons vu des exemples de ces défauts dans la base CERN (défaut 10 et 12). Un dernier groupe dont les défauts 5, 6, 7 font partie et qui déclenchent les mêmes alarmes dans le même ordre. Dans ce cas-là, les méthodes sont incapables de diagnostiquer correctement ces défauts, il faudra alors prendre en considération d'autres éléments, le temps entre deux alarmes, par exemple, ou alors créer des alarmes supplémentaires afin de les rendre diagnosticables.

8.4. Aide à l'opérateur

Dans ce paragraphe, on se concentre sur l'aide que les méthodes peuvent apporter à l'opérateur, au-delà de la proposition d'un ou plusieurs diagnostics.

8.4.1. Connaissances extraites de la méthode vectorielle

8.4.1.1. Prototype

Le prototype vectoriel permet de résumer k exemples d'un même défaut en un unique vecteur binaire. Ce vecteur est une première source d'information sur le défaut, en effet il représente le cœur des alarmes qui le constituent. Ils permettent une représentation simple des défauts. Prenons pour exemple le défaut 5. Traçons les 5 exemples du défaut (à gauche) et le prototype (à droite) extrait de ces 5 défauts sur la même figure. Pour une meilleure représentation, on ne garde que les alarmes déclenchées dans au moins un exemple du défaut.

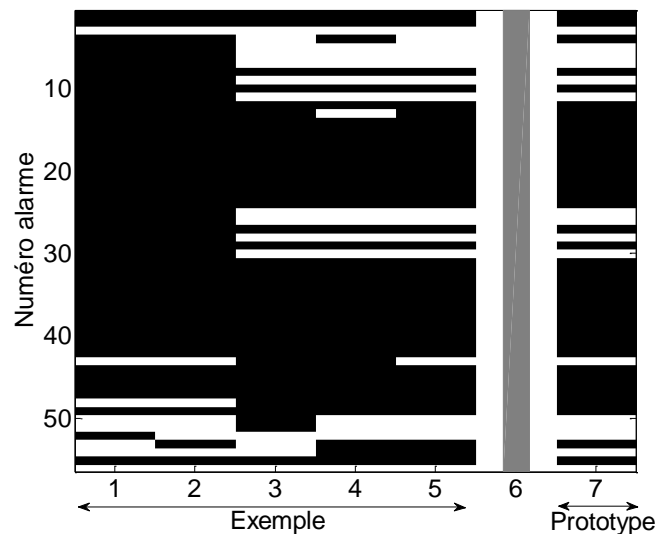


Figure 8-39 Exemple et prototype du défaut 5

8.4.1.2. Poids

Bien que les prototypes soient une manière simple et facilement compréhensible de représenter les défauts, leur nature binaire en fait des éléments limités dans l'aide à l'opération. On a vu que les poids tels que définis dans l'équation 4-8 permettent d'avoir une représentation plus intéressante de l'importance des alarmes pour un défaut. Tout au long du chapitre 6 et du chapitre 8, l'analyse des poids nous a permis de comprendre les défauts représentés et de les comparer. Nous avons vu que les poids calculés dépendent de la base d'apprentissage et notamment du nombre de défauts différents contenus dans la base. Si les poids sont calculés sur toute la base de défauts, on peut dégager des alarmes inutiles car se déclenchant soit avec une probabilité proche de l'aléatoire soit se déclenchant sur l'occurrence de tous les défauts. Il s'agira alors d'étudier ces alarmes et de discuter de leur éventuelle suppression.

Notons enfin que les poids permettent d'analyser l'importance de la présence mais aussi de l'absence d'une alarme dans une liste d'alarmes associée à un défaut. C'est une originalité de la méthode. En effet, l'absence d'une alarme est considérée comme tout autant importante que la présence de l'alarme. L'analyse des poids résume pour l'opérateur quelle alarme doit être présente et quelle alarme doit être absente sur l'occurrence du défaut.

Calculer les poids maximums des alarmes sur l'ensemble des défauts permet de mettre en évidence l'importance des alarmes sur la globalité de la base. Les poids maximums peuvent alors être utilisés pour prioriser les alarmes, on peut imaginer utiliser cette information lorsque l'alarme est affichée à l'opérateur pour focaliser son attention sur les alarmes les plus utiles au diagnostic. On présente un moyen de représenter l'importance des alarmes en le combinant avec une représentation séquentielle dans le paragraphe suivant. La Figure 8-40 montre en dessous les poids maximum de chaque alarme.

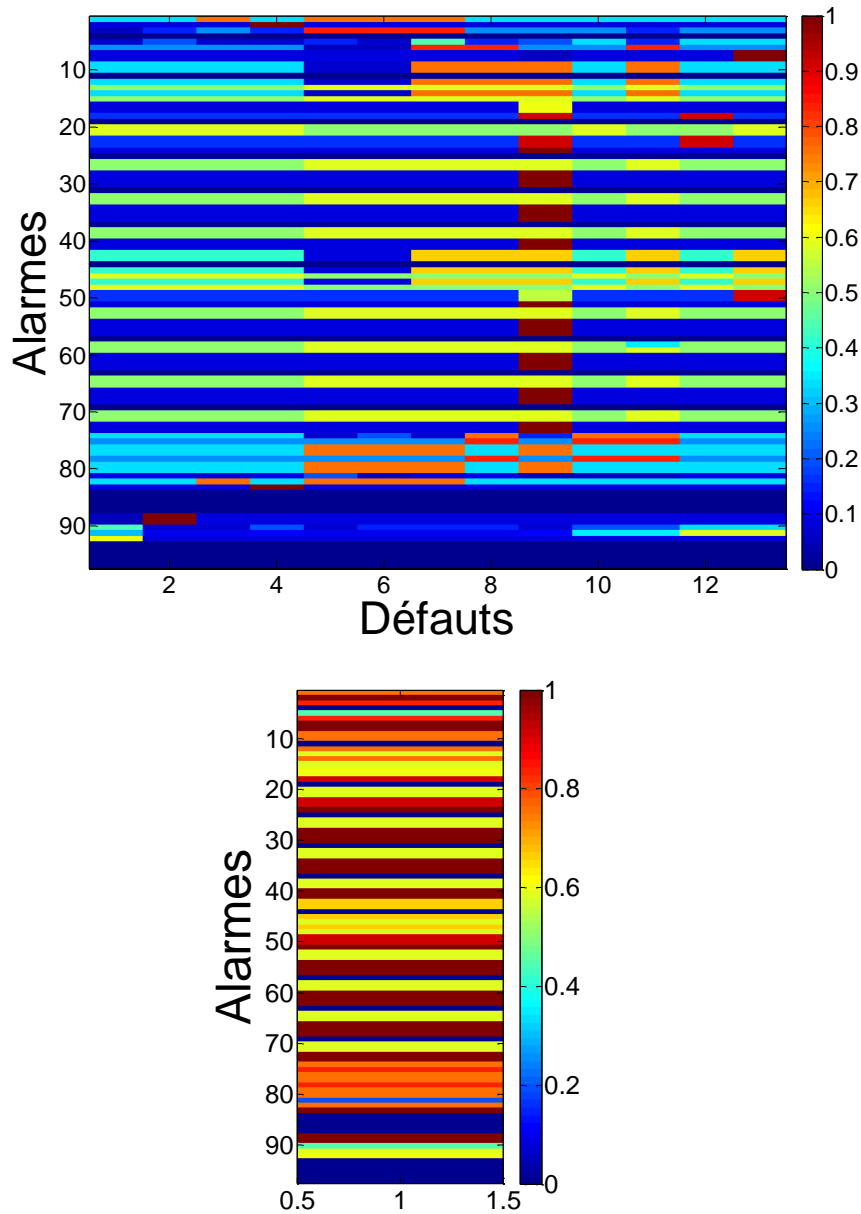


Figure 8-41 Poids max des alarmes

8.4.2. Connaissances extraites des prototypes séquentiels

L'analyse séquentielle permet d'extraire des prototypes séquentiels qui peuvent permettre d'apprendre plus sur le système lors de l'apparition de certains défauts. Il met en relief la séquence d'alarmes qui suit un défaut menant la plupart du temps à l'arrêt du système. Une analyse des prototypes et de la séquence permet de mieux comprendre la propagation du défaut est ainsi développer des méthodes de contrôle afin d'éviter de déclencher cette même séquence d'alarmes et ainsi d'éventuellement prévenir un arrêt du système.

8.4.3. Représentation séquentielle et aide à l'opérateur.

La représentation des défauts à l'aide de séquences permet d'ajouter une information supplémentaire à la simple représentation sous forme de vecteurs. La séquence ajoute l'aspect de causalité entre les alarmes. Représenter les alarmes à l'opérateur sous forme de tracé temporel comme utilisé dans les paragraphes précédents fournit à l'opérateur une représentation informative des alarmes. Les procédés continus étant la majorité du temps une circulation de matières ou de fluides, (comme le système gaz), il est possible en numérotant les alarmes selon leur position géographique dans le système de construire une représentation des alarmes avec une indication de causalité ainsi que de position géographique. En prenant le système gaz, on peut imaginer une représentation identique à celle de la Figure 8-42.

Parallèlement, on peut envisager combiner l'information extraite des poids calculés avec la méthode vectorielle. On propose deux approches, la première en ligne apportant de l'information supplémentaire à l'opérateur lors de l'occurrence d'un défaut et une seconde hors ligne afin d'étudier plus en détail le défaut.

8.4.3.1. Apport en ligne

Chaque avalanche d'alarmes visualisée en représentant chaque alarme se produisant par une couleur et une forme caractéristique de son poids maximum. Cette codification à l'aide de couleur indique ainsi à l'opérateur le degré d'importance de l'alarme dans le système de supervision. Les alarmes se déclenchant fréquemment pour une majorité de défauts sont ainsi mises en évidence.

On propose la codification suivante : noir pour une valeur de poids inférieure à 0.5, bleu pour une valeur comprise entre 0.5 et 0.7 et enfin rouge pour les poids supérieures à 0.7.

La Figure 8-42 donne un aperçu de la représentation proposée couplée à l'information des poids des alarmes. Grâce à la numérotation géographique des alarmes, l'opérateur peut aisément comprendre la propagation des alarmes au sein du procédé. Le code de couleur et de forme lui permet de se focaliser sur les alarmes les plus importantes. Les alarmes sont affichées au fur et à mesure de leur production.

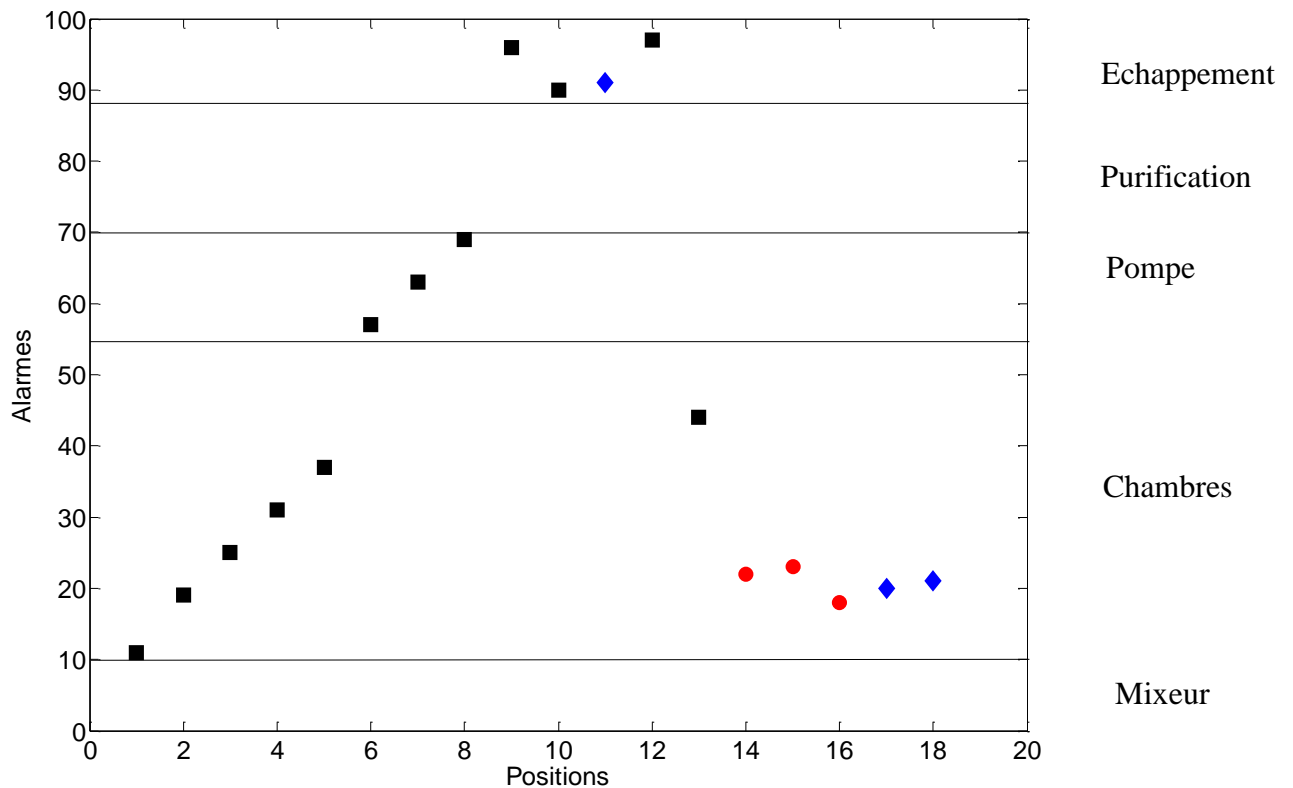


Figure 8-42 Représentation matricielle et aide à l'opérateur

Ainsi, l'analyse vectorielle a mis en évidence que les premières alarmes déclenchées n'étaient pas pertinentes car elles se déclenchaient initialement pour la majorité des défauts. L'opérateur est alerté par le fait que les alarmes se propagent du module échappement vers les modules chambres.

8.4.3.2. Apport hors ligne

L'approche hors ligne est utilisée à la fin de la procédure de diagnostic, pour permettre à l'opérateur de comprendre pourquoi le système a reconnu un défaut particulier. La séquence d'alarmes est alignée avec celle du prototype de défaut reconnu et présentée à l'opérateur sous la même forme graphique que celle proposée en ligne. Les alarmes appartenant au prototype séquentiel sont entourées d'un cercle, afin que l'opérateur puisse juger de la similarité de la séquence produite avec le prototype. Chaque alarme est représentée suivant une couleur et une forme calculée en fonction des poids spécifiques au défaut (en utilisant la stratégie du seul contre tous).

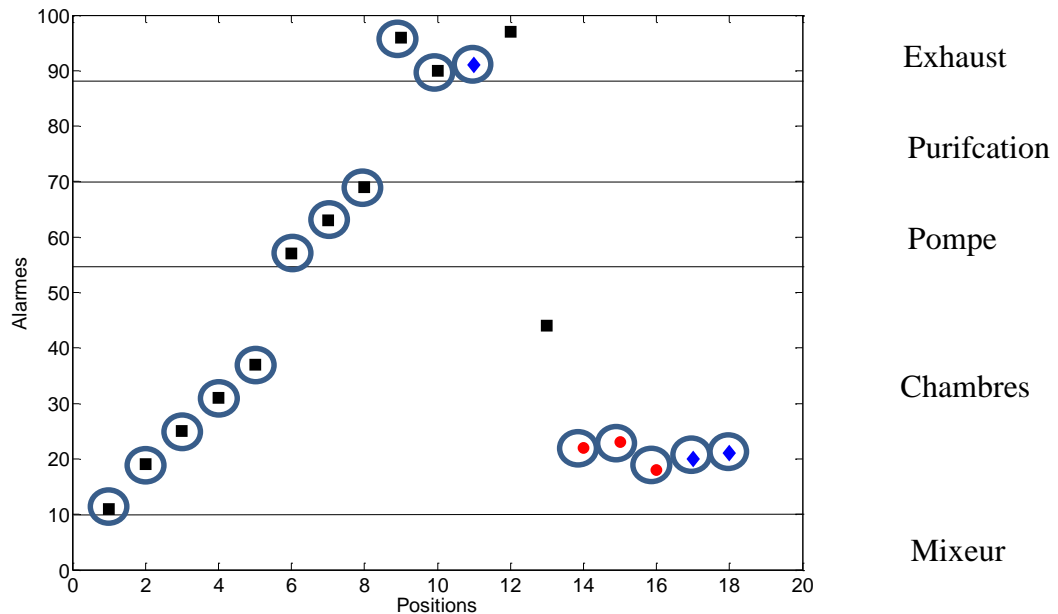


Figure 8-43 Représentation du défaut à identifier après alignement

Dans la Figure 8-43 on met en évidence les alarmes du défaut inconnu qui apparaissent dans le prototype séquentiel. Ici on s'aperçoit que les alarmes apparaissant aux positions 12 et 13 lors de l'analyse hors ligne ne sont plus présentes lorsqu'elles sont comparées avec le prototype. Mais la majorité des autres alarmes sont présentes avec un ordre respecté. On peut en conclure que l'on a à faire à un défaut proche du prototype séquentiel.

8.5. Conclusion

Ce chapitre présente les résultats et les performances de la méthode de diagnostic à l'aide de comparaison de séquences d'alarmes. Le comportement est d'abord analysé à l'aide d'une base évolutive créée artificiellement simulant plusieurs cas de figures qui peuvent être rencontrés dans les systèmes réels. Après une étude de l'influence des différents paramètres de réglage, les performances de Sim_{NWnorm} sont alors confrontées à ceux de la méthode vectorielle. Les avantages et les inconvénients de chaque méthode en fonction de la base sont alors étudiés.

L'utilisation de prototypes séquentiels a un intérêt lorsqu'il n'existe pas d'alarmes réellement discriminantes mais qu'il existe un certain nombre d'alarmes dont l'ordre d'apparition est propre et unique pour chaque défaut, et ce même si le nombre d'alarmes se déclenchant aléatoirement est important. Dans cette configuration, les résultats sont très bons, avec une courbe COR proche de l'idéal. L'ajout de séquences d'alarmes qui se répètent avec un même ordre quel que soit le défaut vient perturber le bon fonctionnement de l'algorithme séquentiel et ceci à travers deux mécanismes. Lors de la construction de prototypes séquentiels, ces alarmes peuvent empêcher l'extraction correcte de ses séquences

représentatives. Ces séquences répétées augmentent également de façon artificielle les similarités entre les séquences et donc perturbent la différenciation entre les défauts. Notons tout de même que cette difficulté peut être surmontée en choisissant une matrice de coût adaptée qui pénaliserait ces alarmes. Après l'analyse des différents comportements de l'algorithme séquentiel sur la base artificielle, les résultats sur la base du CERN ont été étudiés. Une étude de l'influence des paramètres sur les résultats de diagnostic a fait ressortir une certaine stabilité des performances en fonction des paramètres excepté, pour le cas extrême où $\lambda = 1$. Les performances ont été évaluées sur la base du simulateur et comparées à ceux de D_{vp} . Les analyses montrent de bons résultats pour les deux méthodes. Pour un taux de bonnes détections égal à 90%, Sim_{NWnorm} associée à une décision par seuil permet de réduire la liste des défauts possibles à 1 pour 10 défauts sur 13 alors que D_{vp} fournit en moyenne une liste de deux défauts. Par contre, quand le taux de bonnes détections est fixé à 98%, D_{vp} permet une meilleure discrimination entre défauts avec en moyenne 2 défauts proposés quand Sim_{NWNorm} en propose jusqu'à 4. La capacité de D_{vp} à prendre en compte la valeur des alarmes dans les autres défauts lui permet de mieux discriminer certains défauts caractérisés par de nombreuses alarmes communes, à l'exception de quelques alarmes spécifiques. Enfin, la fin de ce chapitre est consacrée à la proposition d'outils utilisant les prototypes séquentiels ainsi que les poids calculés avec la méthode vectorielle comme moyen d'extraire de la connaissance, avec comme objectif l'aide à l'opérateur et à la compréhension du système.

9. Conclusion

Dans ce travail de thèse, nous avons proposé des méthodes de diagnostic utilisant des listes d'alarmes, adaptées aux problématiques de surveillance des procédés du CERN. Une liste d'alarmes est l'ensemble des alarmes déclenchées par le système de contrôle sur l'occurrence d'un défaut, depuis la première alarme jusqu'à la stabilisation du flot d'alarmes. Les systèmes du CERN sont nombreux et très différents les uns des autres. De plus, relativement peu d'exemples en situation de défauts sont disponibles. Ces contraintes excluent l'utilisation de méthodes reposant sur de la connaissance des systèmes (modèles ou bases de règles) ainsi que de méthodes d'apprentissage automatique, trop gourmandes en données. Ainsi, les algorithmes développés dans ce travail reposent sur la comparaison du défaut à diagnostiquer avec des exemples de défauts déjà observés sur le système à l'aide de distances ou de similarités. Le diagnostic est effectué soit à l'aide d'un seuil sur la distance/similarité ou alors par une procédure du plus proche voisin. Deux représentations des données sont étudiées. La première suppose que l'ordre d'apparition d'une alarme a peu d'importance lorsqu'il s'agit de faire un diagnostic sur de tels systèmes et les listes d'alarmes sont transcrites sous forme de vecteurs binaires. La seconde, quant à elle, prend en compte l'ordre d'apparition des alarmes. Les listes d'alarmes sont alors transcrites en séquences symboliques. Ces méthodes ont été testées sur des données extraites d'un simulateur très réaliste d'un procédé du CERN qui a été développé au cours de la première année de la thèse. Le simulateur a demandé un temps important de développement (presque 10 mois). Il nous a semblé néanmoins nécessaire de le réaliser de manière à valider les méthodes proposées sur une base de données possédant les caractéristiques des procédés et des systèmes de contrôle du CERN. Dans ce simulateur, le modèle mathématique du procédé physique est directement relié au vrai système de contrôle du CERN, ce qui nous permet de générer des listes d'alarmes cohérentes avec ce qu'un opérateur analyse en situation de défaut. La difficulté quand on réalise une base de données à partir d'un simulateur est d'arriver à créer de la variabilité, c'est-à-dire générer des exemples différents les uns des autres pour un même défaut. Dans notre cas, les exemples de défauts ont été créés en partant d'états différents pour le système. L'utilisation d'une simple distance de Hamming, dont les performances se sont avérées médiocres, nous a confirmé que les exemples étaient bien différents et que le problème à résoudre n'était donc pas trivial.

9.1. Méthode de diagnostic par comparaison de vecteurs d'alarmes

Dans un premier temps, nous avons considéré que l'ordre d'apparition des alarmes avait peu d'importance et avons proposé une représentation des défauts sous forme vectorielle. Chaque ligne du vecteur de défauts renseigne sur le fait que l'alarme a été déclenchée (ou non) pendant l'avalanche d'alarmes. Un unique vecteur appelé prototype synthétise les différents exemples disponibles sur un même défaut. La comparaison entre le vecteur du défaut à diagnostiquer et le prototype se fait à l'aide d'une distance vectorielle pondérée et normée. L'originalité de la méthode réside dans le calcul des poids. La pondération est un élément important pour différencier les alarmes. Dans la quantité totale

d'alarmes déclenchées lors d'un défaut, peu d'alarmes sont en réalité pertinentes et utiles au diagnostic. Il s'agit alors de mettre en avant ces alarmes. Alors que les méthodes de la littérature proposent des calculs de poids uniquement basées sur les exemples d'un même défaut, nous proposons une approche différente en calculant les poids à partir de tous défauts dont on dispose. L'idée de base de calcul du poids d'une alarme est la suivante : Une alarme est d'autant plus pertinente pour un défaut que sa valeur (0 ou 1) est identique pour un grand nombre d'exemples du défaut. Une alarme est pertinente pour le défaut comparé à un ensemble d'autres défauts si la valeur prise par l'alarme dans le défaut est le contraire de la valeur prise par l'alarme sur l'occurrence des autres défauts. Ces deux affirmations sont traduites sous forme mathématiques par deux termes distincts.

Les poids peuvent donc être calculés en fonction de différents contextes. Celui-ci est défini par le nombre de défauts auquel un défaut est comparé. Un défaut peut être comparé à l'ensemble des défauts connus sur le système, à un ensemble plus restreints de défauts connus pour déclencher des listes d'alarmes similaires, ou deux défauts peuvent être comparés l'un avec l'autre. Ces poids permettent d'extraire de la connaissance du système d'alarmes en identifiant les alarmes réellement pertinentes pour diagnostiquer un défaut. Ils peuvent servir à un premier filtrage permettant d'éliminer les alarmes non informatives.

La distance normée sert de base au diagnostic. Elle peut être seuillée ou utilisée dans un classifieur du type k plus proches voisins. Le calcul des poids en fonction du contexte nous a permis de proposer une stratégie de diagnostic en deux étapes, où le nombre de défauts potentiels est initialement réduit, ce qui permet un re-calcul des poids et un affinage du diagnostic. Quand le diagnostic est réalisé à l'aide de l'algorithme des plus proches voisins (en une ou deux étapes), la méthode ne nécessite aucun paramètre de réglage.

Ces méthodes de diagnostic ont été évaluées dans un premier temps sur une base artificielle, générée à l'aide de générateurs aléatoires, puis sur la base de données du CERN. Elles ont montré des résultats bien supérieurs à ceux obtenus par la distance de Hamming et permettent de diagnostiquer des défauts même lorsque le nombre d'alarmes spécifiques à un défaut est faible.

9.2. Méthode de diagnostic par comparaison de séquences d'alarmes

Dans la seconde partie de ce travail de thèse, l'ordre d'apparition des alarmes est pris en compte. Le formalisme choisi est celui d'une séquence symbolique. Les méthodes développées sont des méthodes inspirées du domaine de la Bio-Informatique, car les séquences de gènes présentent de fortes similitudes avec les séquences d'alarmes. En effet, du fait du caractère non déterministe des procédés physiques, les séquences déclenchées par un même défaut ne seront jamais les mêmes en fonction de l'état du système ou de l'environnement extérieur. Les méthodes de comparaison en Bio-informatique présentent l'avantage d'être tolérante à ces différences. Nous avons donc adapté les méthodes de

comparaison de séquences d'ADN à une problématique de diagnostic de systèmes complexe en définissant un alphabet constitué des alarmes pouvant être déclenchées par le système de contrôle. Des prototypes séquentiels de défauts sont extraits par de l'alignement multiple, en utilisant l'algorithme de Needleman and Wunsch, grâce auquel une séquence représentative du défaut peut être obtenue avec un nombre limité d'exemples par défaut. La comparaison entre séquences de défauts à diagnostiquer et prototype se fait ensuite à l'aide de calcul de similarité entre séquences reposant sur l'alignement de deux séquences. Contrairement à la méthode proposée pour comparer des vecteurs, cette méthode nécessite la définition de plusieurs paramètres : le coût d'un gap (qui représente le coup d'une manipulation sur la séquence) et la matrice de coût de substitution qui mesure le coût de substitution d'une alarme par une autre.

L'algorithme de diagnostic par comparaison de séquences a été ensuite appliqué sur différentes bases créées artificiellement et représentant chacune une situation de défaut différente. La méthode prouve son utilité lorsque les défauts de la base sont constitués d'alarmes respectant le même ordre d'apparition entre les exemples d'un même défaut et à condition que ces alarmes n'apparaissent pas avec le même ordre d'apparition dans les exemples des autres défauts. Les résultats sont bons même lorsqu'il n'existe aucune alarme spécifique aux défauts de la base. La méthode est ensuite appliquée aux données extraites du simulateur du procédé du CERN. Bien qu'elle présente de bons résultats, ils sont en deçà de ceux obtenus par la méthode par comparaison de vecteurs. Ceci s'explique par le fait que, dans la base de défaut du système gaz, ils existent de nombreuses séquences d'alarmes présentent avec le même ordre pour des défauts différents. Alors que la méthode vectorielle est capable de discriminer ces défauts car les poids sont calculés en prenant en compte les autres défauts de la base, la méthode séquentielle construit les prototypes en utilisant uniquement les exemples d'un même défaut. Elle ne peut donc pas isoler les alarmes caractéristiques d'un défaut ie ne se produisant que pour ce défaut. Cependant, nous avons montré qu'il existe des défauts que la méthode séquentielle est capable de mieux diagnostiquer. Il s'agit de défauts où les mêmes alarmes sont déclenchées mais dans un ordre différent.

9.3. Bilan sur les deux méthodes

Nous avons donc proposé dans ce manuscrit deux méthodes de diagnostic originale, qui ont obtenues de bons résultats sur la base de données du CERN et qui ont chacune leur intérêt en fonction du type de défauts auxquels on est confronté.

L'avantage de ces méthodes est qu'elle ne nécessite que peu de données d'apprentissage. Des résultats ont pu être obtenus avec seulement cinq exemples par défauts pour apprendre les prototypes.

Bien que l'information soit représentée de manière différente, toutes les deux utilisent aussi bien le fait qu'une alarme se déclenche sur l'occurrence d'un défaut, que le fait que

l'alarme ne se déclenche pas. Le diagnostic est obtenu grâce aux uns mais aussi aux zéros. Dans la méthode vectorielle, la définition d'une alarme spécifique pour un défaut par comparaison à un ensemble d'autres défauts est donnée de manière identique pour les alarmes se déclenchant que pour celles ne se déclenchant pas. L'information apportée par ces deux situations, traduite par les poids, est considérée comme équivalente. Dans la méthode séquentielle, l'absence d'une alarme dans un prototype va nécessiter l'introduction d'un gap pour pouvoir l'aligner avec la séquence inconnue. De même, la présence d'une alarme dans le prototype va nécessiter l'introduction d'un gap si cette alarme n'est pas présente dans la séquence. La présence ou l'absence d'une alarme va donc être prise en compte par la méthode, de manière équivalente.

Les deux méthodes sont évolutives et vont s'enrichir de nouveaux exemples. L'ajout de nouveaux exemples de défauts déjà connus est facilement pris en compte et nécessite peu de calculs pour obtenir les nouveaux prototypes et poids.

Par contre, les deux méthodes ont montré leur limite pour détecter l'apparition de défauts non encore connus, sur les exemples de la base du CERN. Les résultats sont meilleurs que ceux obtenus par la distance de Hamming, mais restent relativement médiocres. La raison est que les listes d'alarmes générées par le simulateur se ressemblent d'un défaut à l'autre. On observe des groupes de deux ou trois défauts très similaires, qui génèrent en majorité les mêmes alarmes et dans le même ordre. Les méthodes en rejet ne sont pas suffisamment précises pour les distinguer.

Finalement, les deux méthodes sont des méthodes fonctionnant hors ligne. Le diagnostic est effectué à la fin de l'avalanche d'alarmes, une fois la production d'alarmes stabilisée. Le délai de diagnostic peut donc être relativement long, en fonction du temps mis par les alarmes à se propager. Une analyse en temps a été réalisée pour la méthode vectorielle, qui montre que le diagnostic pourrait être obtenu plus tôt, avant la stabilisation des alarmes. En effet, les alarmes se déclenchant à la fin sont des alarmes de propagation, n'ayant plus de lien évident avec l'origine du défaut et influant peu le diagnostic. Cependant, il peut être difficile d'établir un temps T à partir duquel le diagnostic peut être fourni pour un procédé donné. Le problème n'a pas été abordé au cours de cette thèse. Extraction de connaissances et aide à l'opérateur

Au-delà du diagnostic en lui-même réalisé par les deux méthodes, celles-ci permettent toutes les deux d'extraire de la connaissance sur le procédé en défaut. Les poids renseignent sur la pertinence d'une alarme en général et sur la pertinence d'une alarme pour un défaut donné. Ils informent les opérateurs sur les alarmes permettant de discriminer deux défauts proches. Ils peuvent permettre une révision des alarmes présentées en suggérant de les supprimer du système de contrôle si leur poids est faible. Les prototypes séquentiels représentent la séquence d'alarmes caractéristique d'un défaut.

Des méthodes de présentation de l'information pouvant faciliter le travail de diagnostic de l'opérateur ont été proposées. Elles combinent les informations extraites grâce

aux poids des défauts et la représentation séquentielle des alarmes. En effet, nous avons proposé une représentation graphique des listes d'alarmes combinant trois informations : l'ordre d'apparition des alarmes, le lieu de leur apparition ainsi que l'importance de l'alarme dans un but de diagnostic.

9.4. Perspectives

Les méthodes développées dans ce travail ont montrées de bonnes performances sur le simulateur servant à leur validation. Une étude de leur intégration est maintenant en cours. Il existe plusieurs axes de réflexion qui n'ont pas été abordés, comme les moyens de capturer la connaissance sur les systèmes afin de créer la base d'apprentissage. Demander à l'opérateur de renseigner lui-même dans un premier temps le diagnostic afin de remplir la base apparaît être une solution plausible. Cependant il faut réfléchir à l'interface à mettre en place afin de récolter ces informations sans encombrer l'opérateur, déjà surchargé lorsqu'un défaut apparaît. L'autre axe purement technique est le moyen d'intégration d'un tel système dans un système de contrôle industriel.

Nous avons développé dans ce travail de thèse deux méthodes de diagnostic distinctes, qui peuvent en fait être combinées. En effet, nous avons vu que, dans certaines situations, la méthode vectorielle se comporte mieux que la méthode séquentielle et vice et versa. Une étude de l'association des deux méthodes serait un axe intéressant d'amélioration des performances des méthodes. L'objectif serait de les unifier afin de ne pas avoir à choisir entre les deux et augmenter ainsi leurs performances respectives. On pourrait, par exemple, imaginer appliquer les deux algorithmes en parallèle et prendre la décision en fonction des sorties des deux algorithmes à l'aide d'un OU ou d'un ET logique, ou par des moyens plus sophistiqués. Un autre moyen de rapprocher les deux méthodes pourrait consister à adapter le calcul de poids à la méthode séquentielle. En effet, la diagonale de la matrice de coût représente en quelque sorte la valeur de poids des alarmes. Adapter des poids calculés pour des vecteurs à une la méthode séquentielle permettrait d'améliorer les performances de la méthodologie.

Enfin, l'introduction du temps dans le diagnostic n'a pas été analysée. Le diagnostic pourrait-il est posé plus rapidement ? Quelles méthodes permettraient de mieux définir le temps T au bout duquel le diagnostic serait fourni ? Une version en ligne pourrait-elle être imaginée, qui permettrait de fournir un diagnostic dès que le système le peut et non pas au bout d'un temps fixe ? Le temps séparant l'apparition de deux alarmes est aussi une information non utilisée. La méthode séquentielle pourrait-elle être modifiée pour prendre en compte cette information ?

10. Bibliographie

- Agrawal, R., & Srikant, R. (1995). Mining Sequential Patterns. *Proceedings of Conference on Data Engineering (ICDE)*. Taipei, Taiwan.
- Al-Jumah, A., & Arslan, T. (1998). Artificial neural network based multiple fault diagnosis in digital circuits. *Proceedings of ISCAS*, 2, pp. 304-307. Monterey.
- Al-Taani, A. T. (2007). An Expert System for Car Failure Diagnosis. *Proceedings of World Academy of Science, Engineering and Technology*, 1, 476-480.
- Bernard, J., & Duroche, D. (1993). An Expert System for Fault Diagnosis Integrated in Existing SCADA systems. *Proceedings of Power Industry Computer Application Conference*, 9, pp. 313-319. Scottsdale.
- Boufaied, A. (2003). *Contribution à la surveillance distribuée des systèmes à événements discrets complexes*. Université Paul Sabatier : thèse de doctorat.
- Bradru, B. (2010). *Modélisation Simulation et Contrôle des installations cryogéniques du CERN*. Université de Paris-Sud XI: thèse de doctorat.
- Budalakoti, S., Srivastava, A. N., & Otey, M. E. (2009). Anomaly Detection and Diagnosis Algorithms for Discrete Symbol Sequences with Applications to Airline Safety. *IEEE Transaction on Systems, Man, and Cybernetics*, 39(1), 101-113.
- Cabasino, M., Giua, A., Pocci, M., & Seatzu, C. (2007). Identification of Petri Nets from Knowledge of Their Language. *Discrete Event Dynamic Systems*, 17(4), 447-474.
- Chandola, V., Banerjee, A., & Kumar, V. (2012). Anomaly Detection for Discrete Sequences: A Survey. *IEEE Transactions on knowledge and data engineering*, 24(5), 823-839.
- Charkaoui, N. (2005). *Discrimination de classes à occurrences simultanées par des approches de reconnaissance des formes : application à la détection et à la localisation de défaillances sur les véhicules en après-vente*. Université de Compiègne: thèse de doctorat.
- Charkaoui, N., Dubuisson, B., & Ambroise, C. (2005). Pattern recognition method for off-board automotive vehicle failure isolation. *Proceedings of 16th IFAC world Congress*. Prague.
- Charniak, E. (1991). Bayesian Networks without Tears. *AI Magazine*, 12(4), 50-63.
- Charpak, G. (1969). Evolution of some particle detector based on the discharge in gases. *The new astronomy*, 161-180.

- Chen, B., Qiu, Y., Feng, Y., & Tavner, P. (2011). Wind turbine SCADA alarm pattern recognition. *Proceedings of Renewable Power Generation*, (pp. 1-6). Edinburgh.
- Chen, B., Tavner, P., & Feng, Y. (2012). Bayesian Networks for Wind Turbine Fault Diagnosis. *Proceedings of Conference EWEA*. Copenhagen.
- Cheng, Y., Izadi, I., & Chen, T. (2013). Pattern matching of alarm flood sequences by a modified Smith-Watermann algorithm. *Chemical Engineering Research and Design*, 91(6), 1085-1094.
- Cordier, M.-O., & Dousson, C. (2000). Alarm Driven Monitoring on Chronicles. *Proceedings of safeprocess*, (pp. 286-291). Budapest.
- Corpet, F. (1988). Multiple sequence alignment with hierarchical clustering. *Oxford Journals*, 16(22), Pp. 10881-10890.
- Costa, R., Cachulo, N., & Cortez, P. (2009). An Intelligent Alarm Management System for Large-Scale Telecommunication Companies. *EPIA '09 Proceedings of the 14th Portuguese Conference on Artificial*, (pp. 386 - 399). Aveiro.
- de Kleer, J. (1987). *Local Methods for Localizing Faults in Electronic Circuits*. MIT Artificial Lab. Memo No. 394.
- Dousson, C., & Vu Duon, T. (1999). Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. *Proceedings of International Joint Conference on Artificial Intelligence*. Stockholm.
- Duong, T. V. (2001). *Découverte de chroniques à partir de journaux d'alarmes : application à la supervision de réseaux de télécommunications*. Toulouse INPT: Thèse de doctorat en Informatique et télécommunications.
- Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Oxford Journals*, 32(5), 1792-1797.
- Feng, D., & Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25(4), 351-360.
- Folmer, J., & Vogel-Heuser, B. (2012). Computing Dependent Industrial Alarms for Alarm Flood Reduction. *Proceedings of 9th International Multi-Conference on Systems, Signals and Devices (SSD)*, (pp. 1-6). Chemnitz.
- Gardner, R., & Harle, D. (1996). Methods and systems for alarm correlation. *Proceedings of Global Telecommunications Conference*, (pp. 136-140). London.
- Geymayr, J., & Ebecken, N. F. (1995). Fault-Tree Analysis: A Knowledge-Engineering Approach. *IEEE Transactions on Reliability*, 44(1), 37-45.

- Giua, A., & Seatzu, C. (2001). Design of observers/controllers for discrete event systems using Petri nets. *Synthesis and Control of Discrete Event Systems* (pp. 167-182). Springer US.
- Guerraz, B., & Dousson, C. (2004). Chronicles Construction Starting from the Fault Model of the System to Diagnose. *Proceedings of 15th International Workshop on Principles of Diagnosis (DX'04)*, (pp. 51-56). Carcassonne.
- Hashtrudi Zad, S., Kwong, R., & Wonham, W. (1999). Fault diagnosis in finite-state automata and timed discrete-event systems. *Proceedings of 38th IEEE Conference on Decision and Control*, (pp. 81-105). Phoenix.
- Hatonen, K., Klemettinen, M., Mannila, H., Ronkainen, p., & Toivonen, H. (1996). Knowledge Discovery from Telecommunication Network Alarm Databases. *Proceedings of Data Engineering*, (pp. 115-122). New-Orlean.
- Higgins, D., & Sharp, P. M. (1988). CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, 73(1), 237-244.
- Hofmeyr, S. A., Forrest, S., & Somayaji, A. (1998). Intrusion Detection using Sequences of System Calls. *Journal of Computer Security*, 6(3), 151-180.
- Hollifield, B., Oliver, D., Nimmo, I., & Habibi, E. (2008). *HMI Handbook*. Plant Automation Services; 1st edition.
- Hostalkova, E., & Stluka, P. (2010). Alarm rationalization support via. *Proceedings of 19th International Congress of Chemical and Process Engineering*. Prague.
- Huang, Y.-C. (2002). Abductive Reasoning Network Based Diagnosis System for Fault Section Estimation in Power Systems. *IEEE Transactions on Power Delivery*, 17(2), 369-374.
- Hurdle, E., Barlett, L., & Andrews, J. (2005). System fault diagnostics using fault tree analysis. *Proceedings of the 16th Advances in Reliability Technology Symposium (ARTS)*. Loughborough.
- Iserman, R. (2004). Model-Based Fault Detection and diagnosis status and applications. *Proceedings of 16th IFAC Symposium on Automatic Control in Aerospace*. St. Petersburg
- Iserman, R., & Balle, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5), 709-719.
- Jakobson, G., & Weissman, M. (1993). Alarm Correlation. *IEEE Network*, 7(6), 52-59.

- Julisch, K. (2001). Mining alarm clusters to improve alarm handling efficiency. *Proceedings of 17th Annual Computer Security Applications Conference ACSAC*, (pp. 12-21). Orlando.
- Karp, R. M., & Rabin, M. (1987). Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development*, 31(2), 249-260.
- Knuth, D., Morris, J., & Pratt, V. (1987). Fast Pattern Matching in Strings. *SIAM Journal on Computing*, 6(2), 323-350.
- Kondo, R. E., Loures, E. d., & Santos, E. A. (2012). Process Mining for Alarm Rationalization and Fault Patterns Identification. *Proceedings of IEEE 17th Conference on Emerging Technologies & Factory Automation*, (pp. 1-4). Krakow.
- Lambert, M., Riera, B., & Martel, G. (1999). Application of functional analysis techniques to supervisory systems. *Reliability Engineering & System Safety*, 64(2), 209-224.
- Larsson, J. (1994). Hyperfast algorithms for model-based diagnosis. *Computer-Aided Control System Design, 1994. Proceedings of IEEE/IFAC Joint Symposium on*, (pp. 533-538). Tucson.
- Lefebvre, D., & Delherm, C. (2007). Diagnosis of DES With Petri Net Models. *IEEE Transactions on Automation Science and Engineering*, 4(1), 114-118.
- Lefebvre, D., & Leclercq, E. (2011). Stochastic Petri Net Identification for the Fault Detection and Isolation of Discrete Event Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 42(2), 213-225.
- Liao, S.-H. (2005). Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert Systems with Applications*, 28(1), 93-103.
- Lind, M. (1994). Modeling Goals and Function of Complex Industrial Plants. *Applied Artificial Intelligence*, 8, 259-284.
- Liu, J., Lim, K. W., Khuen Ho, W., Chen Tan, K., & Srinivasan (2003). The Intelligent Alarm Management System. *IEEE Software*, 20(2), 66-71.
- Lu, C., Boedihardjo, A. P., & Manalwar, P. (2005). Exploiting Efficient Data Mining Techniques to Enhance Intrusion Detection Systems. *Proceedings of IEEE International Conference on Information Reuse and Integration*, (pp. 512-517). Las Vegas.
- Mannila, H., Toivonen, H., & Verkamo, A. I. (1997). Discovery of Frequent Episodes in Event Sequences. *data mining and knowledge discovery*, 1(3), 259-289.

- Navarro, G., & Baeza-Yates, T. (1999). Very fast and simple approximate string matching. *Information Processing Letters*, 72(1-2), 65-70.
- Needleman, S. B., & Wunsch, C. D. (1970). A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *J.Mol.Biol.*, 48(3), 443-453.
- Negnevitsky, M., & Pavlovsky, V. (2005). Neural Networks Approach to Online Identification of multiple failures of protection systems. *IEEE Transactions on Power Delivery*, 20(2), 588-594.
- Pandalai, D., & Holloway, L. (2000). Template languages for fault monitoring of timed discrete event processes. *IEEE Transactions On Automatic Control*, 45(5), 868-882.
- Pencolé, Y., & Subias, A. (2009). A Chronicle-based Diagnosability Approach for Discrete Timed-event Systems: Application to Web-Services. *Journal of Universal Computer Science*, 15(17), 3246-3272.
- Qian, Y., Xiuxi, L., Yanrong, J., & Yanqin, W. (2003). An expert system for real-time fault diagnosis of complex chemical processes. *Expert Systems with Applications*, 24(4), 425-432.
- RAM. (1988). *Reliability, availability, and maintainability dictionary* (éd. Tracy P. Omdahl). ASQC quality press.
- Ramírez-Treviño, A., Ruiz-Beltrán, E., Rivera-Rangel, I., & López-Mellado, E. (2007). Online Fault Diagnosis of Discrete Event Systems. A Petri Net-Based Approach. *IEEE Transactions on Automation Science and Engineering*, 4(1), 31-39.
- Reiter, R. (1987). A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 57-95.
- Roth, M., Lesage, J., & Litz, L. (2009). A residual inspired approach for fault localisation. *Proceedings of 2nd Ifac workshop on dependable control discrete event systems*. Bari.
- Salah S., G. M.-F.-V. (2013). A model-based survey of alert correlation techniques. *Computer Networks*, 57(5), 1289-1317.
- Sampath. M., Sengupta R., Lafortune S., Sinnamohideen & Tenektzis D (1995). Diagnosability of discret event systems. *IEEE Transaction on automatic control*, vol 40., n°9.
- Smith, T. F., & Waterman, M. (1981). Comparison of Biosequences. *Advances in applied mathematics*, 2, 482-489.

- Srikant, R., & Agrawal, R. (1996). Mining Sequential Patterns: Generalizations and Performance Improvement. *Proceedings of Fifth Int'l Conference on Extending Database Technology (EDBT)*. Avignon.
- Sun, H., Guo, Q., Zhang, Y., & Zhang, B. (2012). An online intelligent alarm-processing system based on abductive reasoning network. *Proceedings of IEEE Power and Energy Society General Meeting*, 30(4), 1-5. San Diego
- Taisne, J. (2006). Intelligent Alarm Processor based on Chronicle Recognition for Transmission and Distribution System. *Proceedings of Power Systems Conference and Exposition*. Atlanta.
- Tan, J., A., P., Kirschen, D., & J. Goody, a. J. (2000). An Expert System for the Back-up Protection of a Transmission Network. *Proceedings of IEEE Transactions on Power Delivery*, 15(2), 508-514.
- Tuszynski, J., Larsson, J. E., Nihlwing, C., Öhman, B., & Calzada, A. (2002). A Pilot Project on Alarm Reduction and Presentation Based on Multilevel. *Proceedings of Enlarged Halden Programme Group Meeting*. Storefjell.
- Ushio, T., Onishi, I., & Okuda, K. (1998). Fault Detection Based on Petri Net Models with Faulty Behaviors. *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, (pp. 113-118). San Diego.
- Wu, S., & Manber, U. (1992). AGREP- A Fast Approximate Pattern-Matching Tool. *Proceedings of USENIX Technical Conference*, (pp. 153-162). San Francisco.
- Yamaguchi, T., Inagaki, S., & Suzuki, T. (2012). Data based Construction of Bayesian Network for Fault Diagnosis of Event-Driven Systems. *Proceedings of 8th IEEE International Conference on Automation Science and Engineering*, (pp. 20-24). Seoul.
- Yamanishi, K. & Maruyama, Y. (2005). Dynamic syslog mining for network failure monitoring. *In Proceedings of the 11th ACM SIGKDD, Proceedings of International Conference on Knowledge Discovery and Data Mining*, (pp. 499-508). Chicago.
- Yang, F., Shah, S., Xiao, D., & Chen, T. (2012). Improved correlation analysis and visualization of industrial alarm data. *ISA Transactions*, 51(4), 499-506.
- Yemini, S., Kliger, S., Mozes, E., & Yemini, Y. (1996). High speed and robust event correlation. *IEEE Communications Magazine*, 34(5), 82-90.
- Ying, J., Kirubarajan, T., Pattipati, K., & Patterson-Hine, A. (2000). A hidden Markov model-based algorithm for fault diagnosis with partial and imperfect tests. *IEEE*

Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,
30(4), 463-473.

Zwingelstein, G. (1995). *Diagnostic des défaillances*. Editions Hermès.

11. Annexe

A. LES HYPOTHÈSES

Dans la modélisation certaines hypothèses simplificatrices sont prises pour réduire la complexité du modèle, les hypothèses possibles sont les suivantes :

Hypothèse isotherme : Un processus isotherme est un processus qui a lieu à température constante et uniforme.

Hypothèse isobare : Un processus isobare est un processus dans lequel la pression est constante et uniforme.

Hypothèse isochore : Un processus isochore est un processus où le volume est constant. Cette hypothèse sera considérée comme vraie pour tous les modèles développés dans la suite. On considérera qu'il n'y pas de déformation des composants durant les transformations.

Mélange homogène : On considérera dans tous les modèles que les mélanges de gaz sont des mélanges homogènes c'est-à-dire que la concentration des gaz dans un composant est la même en tous point du composant.

B. BIBLIOTHÈQUE DE COMPOSANTS GAZ

a) LES FONCTIONS GAZ

On définit dans un premier temps les fonctions utiles est communes qui permettent de calculer les différentes propriétés des gaz mis en jeu dans le système gaz :

b) LOI DES GAZ PARFAITS

Le gaz parfait est un modèle thermodynamique permettant de représenter le comportement des gaz réels à faible pression. Le modèle suppose que les 3 grandeurs caractéristiques du gaz : la pression P , la température T et le volume V sont indépendantes de la nature du gaz, il repose sur le postulat qu'à faible pression les interactions électrostatiques entre les molécules composant le gaz sont négligeables.

Les relations entre les trois grandes grandeurs physiques sont exprimées selon la loi des gaz parfait définie de la manière qui suit :

$$P.V = n.R.T$$

Avec

n = la quantité de matière exprimée en mol

R la constante des gaz parfait de valeur $8,314 \text{ J.mol}^{-1}.\text{K}^{-1}$

P La pression d'un gaz en pascal Pa

T La température du gaz en Kelvin K

V le volume du gaz en m^3

La loi peut aussi être écrite de la manière qui suit :

$$P.V = n.R.T$$

$$P = \frac{n}{V} . R.T$$

$$P = \rho . R.T$$

Avec

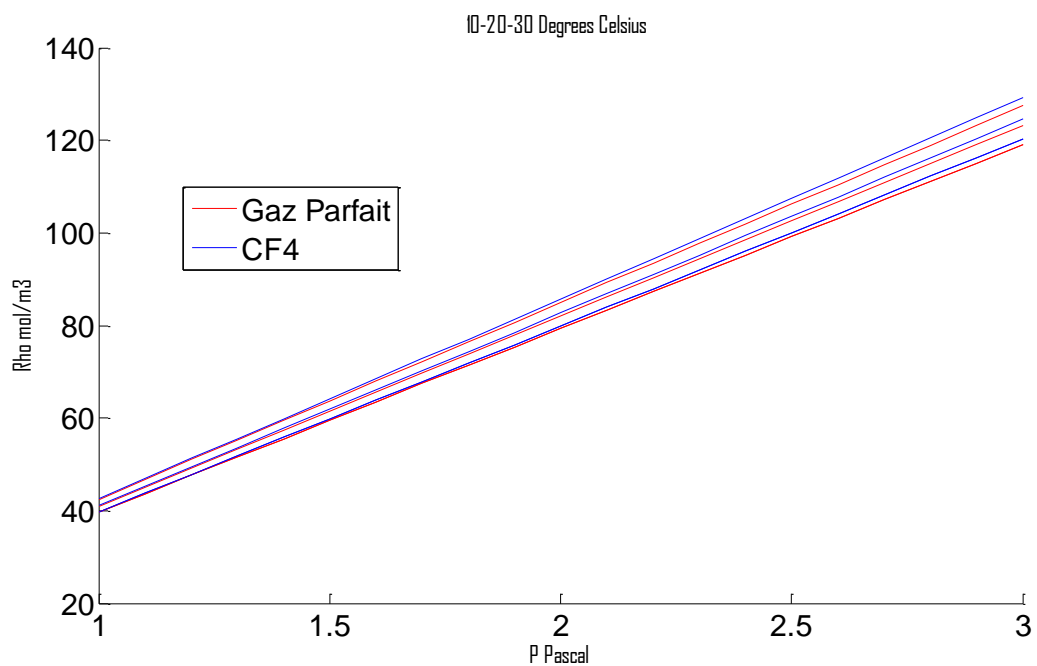
ρ la masse volumique du gaz exprimée en mol/m³

Afin de vérifier que les composés utilisés suivent la loi des gaz parfait on se propose de tracer la courbe de la densité de gaz en fonction de la pression du gaz dans des conditions isotherme c'est à dire à température constante On tracera la courbe pour la loi des gaz parfaits ainsi que pour les valeurs réelles. On choisira trois isothermes (T=10°,20°,30°), et l'on fera varier la valeur de P entre 1bar et 3bar qui correspondent aux valeurs de fonctionnement du système gaz.

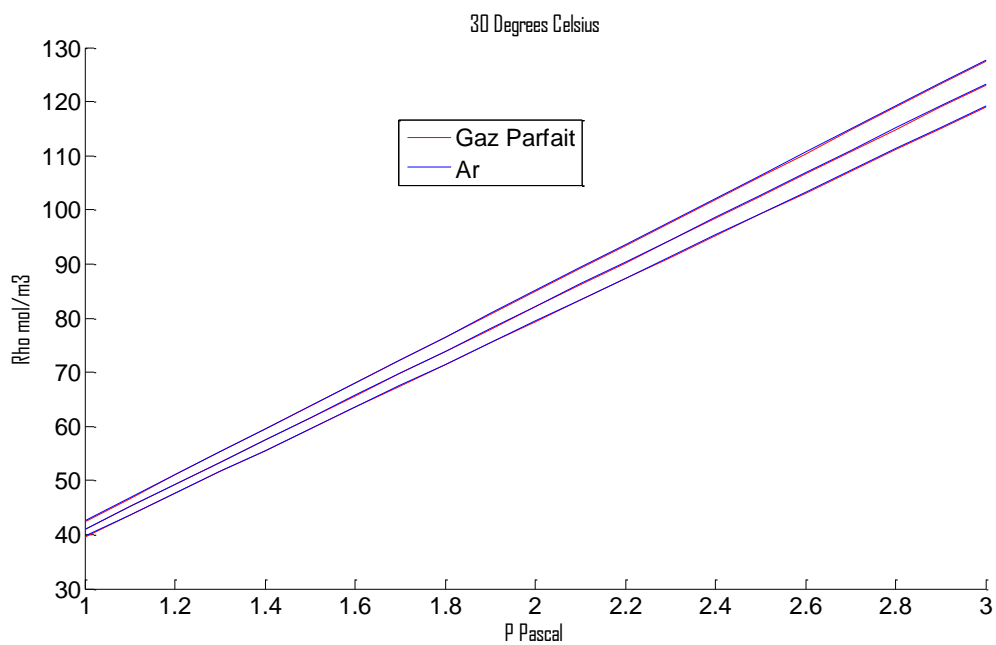
On comparera les deux courbes et calculera l'erreur relative moyenne de la loi des gaz parfaits. La valeur de l'erreur relative moyenne sera obtenue en calculant la moyenne de l'écart relatif sur tous les points de calculs selon la formule qui suit :

$$\epsilon_{relatif} = \frac{(\rho_{gazparfait} - \rho_{réelle})}{\rho_{réelle}} * 100$$

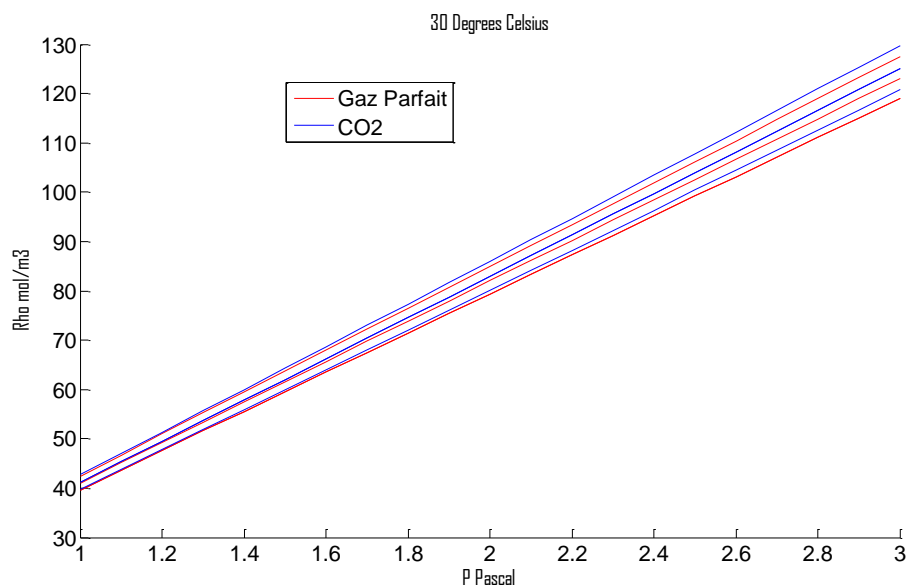
Annexe



T ° Celsius	10°	20°	30°
Erreur Relative %	0.699	0.792	0.898



T ° Celsius	10°	20°	30°
Erreur Relative %	0.1596	0.1362	0.1165



T ° Celsius	10°	20°	30°
Erreur Relative %	1.2027	1.0699	0.9553

L'analyse des 3 courbes et leurs comparaisons avec la loi des gaz parfaits nous montrent une forte similarité de comportement entre elles avec une erreur relative faible en dessous de 1% pour l'Argon et le CF₄, et autour de 1% pour le CO₂. Ceci nous permet d'en déduire que les 3 gaz utilisés dans le gaz système peuvent être considérés comme des gaz parfaits dans les conditions d'utilisations des systèmes gaz. L'hypothèse des gaz parfaits peut être retenue pour les gaz utilisés dans le cadre du système gaz.

c) FONCTION DE MASSE MOLÉCULAIRE :

La masse moléculaire, est la masse d'une molécule qui compose le gaz. Elle correspond à la somme de la masse de chaque atome composant la molécule, elle est obtenue par l'addition de la masse atomique de chaque atome de la molécule multiplié par leur indice numérique de la formule brute :

Soit une molécule X_i de formule brute

$$X_i = \sum_{j=1}^N \sigma_i^j \cdot X_i^j$$

Avec

$$\sigma_i^j \in \mathbb{N}$$

X_i^j l'atome numéro j composant la molécule i

σ_i^j son indice numérique

Soit la formule de calcul de masse moléculaire :

$$MW(X_i^j) = \sum_j \sigma_i^j * mw(X_i^j)$$

Avec

$mw(X_i^j)$ la masse de l'atome X_i^j

Pour les gaz utilisés dans le système modélisé nous avons les résultats suivant :

$$H_2O = 18.015 \text{ kg} \setminus \text{mol} \quad O_2 = 31.9988 \text{ kg} \setminus \text{mol} \quad N_2 = 28.0134 \text{ kg} \setminus \text{mol}$$

$$CO_2 = 44.0098 \text{ kg} \setminus \text{mol} \quad CF_4 = 88.01 \text{ kg} \setminus \text{mol} \quad Ar = 39.948 \text{ kg} \setminus \text{mol}$$

Dans la suite on considérera les molécules dans leurs totalités on restreindra la notation d'une molécule de la manière qui suit X_i .

d) FONCTION DE CONCENTRATION MASSIQUE

Cette fonction calcule le pourcentage en masse de chaque gaz qui compose le mélange de gaz :

Soit un mélange gazeux de masse total M_{tot} et M_i la masse du gaz i dans le mélange, on définit la fraction massique :

$$C_i = \frac{M_i}{M_{tot}}$$

Avec

$$\sum C_i = 1$$

Cette valeur est sans dimension étant donné qu'il s'agit d'un ratio entre les masses des différents gaz composant le mélange de gaz.

e) FONCTION DE FRACTION MOLAIRE

La fraction molaire x_i représente le ratio entre le nombre de molécules du gaz i noté n_i sur le nombre total des molécules n_{tot} .

$$x_i = \frac{n_i}{n_{tot}}$$

Avec

$$\sum x_i = 1$$

Relation entre fraction massique et fraction molaire :

$$x_i = C_i * \frac{M_{moy}}{M_i}$$

Avec

M_{moy} la masse molaire moyenne du mélange de gaz et M_i la masse molaire de la molécule i

Avec

$$x_i = \frac{C_i}{MW(X_i)} * \frac{1}{\sum \left(\frac{C_j}{MW(X_j)} \right)}$$

f) MASSE MOLAIRE D'UN MÉLANGE DE GAZ

La masse molaire d'un mélange d'un gaz est égale à la masse molaire de chaque gaz composant le mélange multipliée par la fraction molaire :

$$MW_{mix}(X_i) = \sum_j x_j * MW(X_j)$$

g) MASSE VOLUMIQUE D'UN GAZ

La masse volumique d'un gaz est le rapport de la masse de ce gaz sur le volume dans lequel il est contenu. On la définit de façon générale par la formule qui suit

$$\rho = \frac{m}{V}$$

Avec ρ la masse volumique du gaz et V le volume dans lequel il est contenu.

Il est cependant plus judicieux d'exprimer la masse volumique en fonction de la pression du gaz une valeur calculée sur l'ensemble des éléments du simulateur.

En partant de l'équation précédente et de la loi des gaz parfaits $P.V = n.R.T$ en remplaçant V par son expression dans la loi des gaz parfait on obtient :

$$\rho = \frac{m}{\frac{nRT}{P}} = P \cdot \frac{m}{nRT}$$

De plus on reconnaît $MW = \frac{m}{n}$ la masse molaire du gaz. On a donc comme équation finale ;

$$\rho = P \cdot \frac{MW}{R \cdot T}$$

Avec

R la constante des gaz parfaits

MW La masse molaire du gaz

T La température du gaz

Pour un mélange de gaz il suffit de remplacer MW par MW_{mix} .

h) VISCOSITÉ

La viscosité peut être définie comme la résistance à l'écoulement du gaz. La densité d'un gaz augmente avec sa température augmentant ainsi sa résistance à l'écoulement. Afin de calculer la densité d'un gaz on se basera sur une approximation polynomiale de la valeur de la viscosité dans la plage de fonctionnement du système gaz. La viscosité d'un gaz sera alors calculée de la manière qui suit :

$$Visc[j] = \alpha_j + \beta_j * T + \gamma_j * T^2$$

Avec T exprimée en °C

$$Visc \text{ en } Pa.s = \frac{Kg}{m.s}$$

C. MODÉLISATION DES COMPOSANT : BIBLIOTHÈQUE GAZ

a) BUFFER

Les buffers sont l'un des principaux éléments capacitifs ils consistent en un gros volume permettant le stockage de gaz. On considère ici la modélisation d'un buffer de forme cylindrique. Il possède un port d'entrée de type gaz et une sortie de type gaz . Il s'agit d'un composant passif sans aucun élément de contrôle. Le modèle repose principalement sur la loi des gaz parfaits. Le volume du buffer est considéré comme constant (hypothèse isochore), et les échanges de température avec l'extérieur sont considérée comme nulles.

Déclaration des variables :

V Volume du buffer en m^3

T Température initiale du gaz dans le buffer en Kelvin

$C[j]$ Fraction massique des gaz

P Pression initiale du buffer Pascal

m Masse totale de gaz dans le buffer en kg

MW Masse molaire du mélange de gaz

Équation différentielle :

Equilibre total des masses

$$\dot{m} = W_{in} - W_{out}$$

Equilibre des masses pour chaque composant

$$\dot{C}[j] = W_{in} * \frac{C_{in}[j] - C[j]}{m}$$
$$1 = \sum C[j]$$

Pression du gaz :

- Calcul de la masse moléculaire

$$MW = MW_{mix}(gaz_{mix}, C)$$

- Loi des gaz parfaits peut s'exprimer de la manière suivante :

$$P = m \cdot R \cdot \frac{T}{MW \cdot V}$$

- Soit la variation de pression définie par :

$$\dot{P} = \dot{m} \cdot R \cdot \frac{T}{MW \cdot V}$$

Équation des ports :

$$P_{out} = P$$

$$P_{in} = P$$

$$T_{out} = T$$

b) PURIFIER

Le purifier est composé d'un gros volume où se trouve à l'intérieur des charbons actifs ayant pour but de capturer les molécules indésirables dans le circuit de gaz que sont les molécules de H_2O et O_2 contenues dans le gaz. Quant aux molécules de N_2 elles ne sont pas filtrées par le purifier est continues leur « route » dans le système. C'est d'ailleurs pour cette raison que du gaz est périodiquement rejeté du système gaz et du mélange de gaz est injecté dans le système afin de garder le taux de N_2 faible dans le circuit. Le purifier sera modélisé par un volume V . On considérera que la totalité des molécules H_2O et O_2 sont absorbées par le purifier.

Déclaration des variables :

V Volume du purifier m^3

T Température initiale du gaz dans le Purifier en Kelvin

$C[j]$ Fraction massique des gaz

P Pression initiale du Purifier en *bar*

m Masse totale de gaz dans le buffer en *kg*

MW Masse molaire du mélange de gaz

Equation du modèle

- Equilibre de la matière

$$\dot{m} = w_{in} - w_{H_2O} - w_{O_2}$$

- Purification :

On définit la réaction de purification des différents éléments c-à-d l'absorption des molécules de O_2 et H_2O

$$\text{Reac}[\text{Ar}] = 0$$

$$\text{Reac}[\text{CF}_4] = 0$$

$$\text{Reac}[\text{CO}_2] = 0$$

$$\text{Reac}[\text{N}_2] = 0$$

$$\text{Reac}[\text{H}_2\text{O}] = -w_{H_2O}$$

$$\text{Reac}[\text{O}_2] = -w_{O_2}$$

- Equilibre de matière pour chaque composant

$$\dot{C}[j] = \frac{w_{in} * (C_{in}[j] - C[j]) + Reac[j]}{m}$$

- Pression des gaz

$$MW = MW_{mix}(gaz_{mix}, C)$$

$$\dot{P} = m \cdot R \cdot \frac{T}{MW \cdot V} - P$$

Equation des ports

$$\begin{aligned} P_{in} &= P_{out} = P \\ T_{in} &= T_{out} = T \end{aligned}$$

c) CHAMBRE DES DÉTECTEURS

La chambre des détecteurs est le lieu où a lieu la détection des particules, il s'agit également d'un volume, relativement fragile qui doit être maintenue à une pression proche de la pression atmosphérique afin d'éviter tout éclatement de cette chambre. Les parois de ces chambres sont poreuses est un phénomène de diffusion de gaz à travers les parois de la chambre a lieu. La diffusion à travers une paroi où diffusion membranaire est un phénomène physique qui a lieu lorsque deux gaz de concentrations différentes se trouvent séparées par une membrane, se sont enfaite les molécules de gaz qui passent à travers les micros espaces dans la matière des parois. Le phénomène de diffusion tend à équilibrer les concentrations des gaz de chaque côté de la paroi. La vitesse de diffusion dépend donc de la paroi (plus ou moins poreuse), la concentration des différents gaz de part et d'autre de la paroi, et de la taille des molécules de gaz. La vitesse de diffusion est d'autant plus grande que la paroi présente des « trous » grand, que les différences de concentrations des gaz de part et d'autre de la paroi est importante ainsi. La taille des molécules quant à elle joue un rôle inverse à la vitesse de diffusion à travers les parois.

Le phénomène de diffusion peut alors se résumer par la loi de Fick :

Pour un gaz on note J la quantité de matière passant de la chambre à l'air est décrite par la relation suivante :

$$J_{mix}[j] = -\lambda_j \cdot (x_i[j] \cdot P - x_i[j] \cdot P_{air})$$

avec λ la constante de diffusion du gaz j . Dépendant de la géométrie de la chambre ainsi que de taille de la molécule exprimée en $Kg \cdot bar^{-1} \cdot s^{-1}$

J_{mix} exprimée en $Kg \cdot bar^{-1} \cdot S^{-1}$

Soit

$$J_{tot} = \sum_{j=1}^N J_{mix}[j]$$

Avec J_{tot} la quantité de matière échanger entre le détecteur et l'air. Si la quantité est positive alors la masse des molécules introduites dans la chambre due à la diffusion est plus importante que la masse introduite dans la chambre, inversement si J_{tot} est négative.

- La variation de masse s'écrit alors

$$\dot{m} = w_{in} - w_{out} + J_{tot}$$

- La concentration des éléments se calcule alors de la manière qui suit :

$$\dot{C}[j] = \frac{w_{in} * (C_{in}[j] - C[j]) + j_{mix}[j]}{m}$$

La Pression à l'intérieur des chambres est quant à elle calculée à l'aide de la loi des gaz parfaits déjà citée précédemment.

d) TUYAU SANS PERTE DE CHARGE

Un tuyau est un composant passif dans lequel le gaz circule. Il peut être également considéré comme un élément de volume si la longueur du tuyau n'implique pas une perte de charge significative dans le système, c'est-à-dire si sa longueur est faible. Son modèle sera alors fortement similaire à celui du buffer. Il s'agit en fait d'un modèle simplifié du buffer. Notamment en ce qui concerne les différentes concentrations en gaz, compte tenu du volume réduit des tuyaux la concentration C du gaz contenue dans le tuyau sera considérée comme égale à la concentration d'entrée des différents gaz. Cette simplification est nécessaire pour des soucis de temps de calcul et de réversibilité de l'équation qui implique elle aussi un temps de calcul réduit.

L'équation d'équilibre des masses devient alors :

$$C[j] = C_{in}[j] \forall j$$

e) TUYAU AVEC PERTE DE CHARGE

Lorsque le tuyau induit une perte de charge non négligeable dans le système, il peut alors être considéré comme un élément résistif avec une perte de charge significative. Le débit au travers du tuyau peut alors être calculé suivant l'équation de Darcy-Weisbach pour une perte de charge régulière dans un tuyau :

$$\Delta P = P_{out} - P_{in} = \Lambda * \frac{L}{D} * \frac{\rho * v^2}{2.10^5}$$

Avec

L longueur du tuyau en m

D le diamètre du tuyau en m

v La vitesse du gaz exprimée en $m.s^{-1}$ avec $v = \frac{F}{a}$ où a est la section du tuyau

Le coefficient de frottement Λ appelé aussi coefficient de Darcy-Weischbach ou encore coefficient de perte de charge peut être déterminé selon le type d'écoulement à l'aide du nombre de Reynolds.

Pour un écoulement laminaire :

$$\Lambda = \frac{64}{Re}$$

Pour un écoulement turbulent :

$$\frac{1}{\sqrt{\Lambda}} = 2. \ln(Re. \sqrt{\Lambda}) - 0.8$$

Le nombre de Reynolds étant défini par :

$$Re = v. \frac{L}{visc}$$

Avec $visc$ la viscosité du gaz exprimé en $Pa.s$

f) PERTE DE CHARGE SINGULIÈRE

Certain tuyau présentent des géométries particulières qui peuvent provoquer une perte de charge singulière, d'autres éléments également peuvent présenter des profils de perte de charge singulière tel un filtre par exemple dans le circuit du gaz. La perte de charge singulière s'exprime de la manière suivante :

$$\Delta P = K. \rho. \frac{v^2}{2}. 10^{-5}$$

Le coefficient K est relativement complexe à déterminer en général. Il peut s'obtenir de façon expérimentale en mesurant les débits et la perte de charge pour un gaz donné sur le système.

g) MIXER

Le mixer est un gros volume dans lequel les gaz purs sont mélangés afin de créer le mélange de gaz qui sera envoyé dans les chambres. Le système modélisé traitant 3 gaz, le mixer sera un mixer avec 3 ports d'entrées (un pour l'Ar, un autre pour le CO2 et un dernier pour le CF4) ainsi qu'une sortie pour le mélange. Le mélange est fait de façon passive c'est-à-dire qu'il n'existe pas d'élément en mouvement effectuant le mélange, le mixage se fait uniquement par diffusion du gaz, il sera considéré comme instantanée et homogène dans ce modèle.

Il reprend en grande partie le modèle du buffer à la différence du nombre de ports d'entrées.

Déclaration des variables :

T Température initiale du gaz dans le buffer en Kelvin

$C[j]$ Fraction massique des gaz

P Pression initiale du buffer Pascal

m Masse totale de gaz dans le buffer en kg

MW Masse molaire du mélange de gaz

Équation différentielle :

- Equilibre de total des masses

$$\dot{m} = (W_{in1} + W_{in2} + W_{in3}) - W_{out}$$

- Equilibre des masses pour chaque composant
-

$$\dot{C}[j] = W_{in} * \left(\frac{C_{in1}[j] - C[j]}{m} + \frac{C_{in2}[j] - C[j]}{m} + \frac{C_{in3}[j] - C[j]}{m} \right) \quad \forall j$$

$$1 = \sum C[j]$$

- Pression du gaz :
- Calcul de la masse moléculaire

$$MW = MW_{mix}(gaz_{mix}, C)$$

- Loi des gaz parfaits peut s'exprimer de la manière suivante :

$$\dot{P} = m \cdot R \cdot \frac{T}{MW \cdot V} - P$$

Équation des ports

$$P_{out} = P$$

$$P_{in} = P$$

$$T_{out} = T$$

h) LES COMPOSANTS RÉSISTIFS

Les composants résistifs sont des composants capables de calculer des débits, leurs conditions aux limites sont les pressions en amont en aval ainsi que les concentrations massiques du gaz au travers des composants.

i) VANNE

La vanne permet de contrôler le débit du gaz dans un circuit, il en existe de nombreux types, elles peuvent contenir une ou plusieurs entrées et une ou plusieurs sorties. De plus elles diffèrent par la manière dont elles contrôlent le débit. La vanne ON/OFF par exemple contrôle le débit de manière binaire elle ne possède que deux positions, la position « ouvert » le débit est alors maximum et la position « fermée » le débit est alors nul. Il existe aussi des vannes analogiques qui peuvent prendre des valeurs d'ouvertures comprises entre 0 et 100% de leur ouverture elles permettent de contrôler avec plus de précision le débit notamment lorsqu'elles sont couplées à un contrôleur de type PID par exemple. Leur méthode est fortement similaire on présentera donc dans un premier temps le modèle partagé par tous les types de vannes on entrera dans le détail de chacune d'elle dans un second temps.

Hypothèse d'écoulement incompressible :

Un écoulement incompressible est un écoulement où il n'y a pas d'accumulation de matière. Un écoulement peut être considéré comme incompressible si le nombre de Mach est inférieur à 0.3. Le nombre de Mach est défini de la manière suivante $M_a = \frac{v}{a}$, où v la vitesse de l'écoulement et a la vitesse du son dans ce même gaz. La Vitesse du gaz est exprimée en $m \setminus s$ dans un tuyau suivant la formule qui suit :

$$v = \frac{F}{A}$$

Où

F est le débit volumique en $m^3 \setminus s$

A La section d'écoulement m^2

En régime normal de fonctionnement le nombre de mach maximum est obtenu pour un débit de gaz est de $8 \text{ Nm}^3 / \text{h}$, et une section d'écoulement de $A = 2. \pi. 0.014^2$. Le nombre de *Mach* est alors égale au maximum de 0.0054 \ll 0.33, soit 61fois inférieur au nombre de mach maximum pour considérer l'écoulement comme incompressible.

Ports :

Port_gaz : f_{in}

Port_gaz : f_{Out}

Équation différentielle d'une vanne :

- Conservation des masses :

$$W_{out} = W_{In}$$

- Conservation de chaque composant :

$$C_{in}^i = C_{Out}^i$$

- Hypothèse isotherme :

$$T_{in} = T_{out}$$

- Pour un écoulement incompressible la loi qui régit le débit d'un fluide au travers d'une vanne est

$$F = K_v \cdot y \cdot \sqrt{\frac{(P_{In} - P_{Out})}{\rho}}$$

Avec F débit en m^3/h

ρ La masse volumique en Kg/dm^3

K_v Coefficient caractéristique de la vanne

P_{In} et P_{Out} en bar

y ordre d'ouverture de la vanne compris en 0 et 1

- En fonction du débit massique W :

$$W = K_v \cdot y \cdot \sqrt{\rho \cdot (P_{In} - P_{Out})}$$

Avec W en Kg/h

Avec les unités utilisées dans la librairie on obtient :

$$W = (K_v \cdot y \cdot \sqrt{\rho \cdot 10^{-3} \cdot (P_{In} - P_{Out})}) \cdot \frac{1}{3600}$$

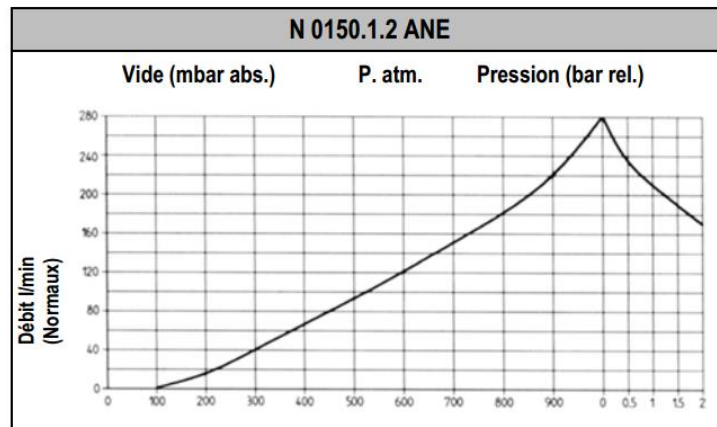
Avec W le débit massique en Kg/s

ρ La masse volumique en Kg/m³

P en bar

j) POMPE

La pompe est le seul composant de la bibliothèque qui fournit un travail au système gaz, elle transforme de l'énergie électrique en énergie mécanique. Une pompe permet d'aspirer et de refouler du fluide, du gaz dans notre cas. Afin de modéliser la pompe notamment le débit de gaz à travers la pompe, on s'appuiera sur les données techniques délivrées par le constructeur. Le constructeur donne les caractéristiques de débit de la pompe en fonction de la pression d'entrée en *bar.rel*. Cette unité de pression correspond à la pression du gaz en bar à laquelle l'on a soustrait une atmosphère qui équivaut à 1023Pa. Les débits sont donnés en normaux litres par heure c'est-à-dire pour une pression de 101 325Pa et une température de 0 degré Celsius.



On peut décomposer le profil de débit en deux phases dans le graphique :

- La première entre 0 *mbar.rel* et 100*mbar.rel*
- La deuxième entre 100*mbar.rel* et 1*bar*
- La troisième entre 1*bar* et 3*bar*

Chaque partie de la courbe peut être approximée par un polynôme afin de modéliser le débit de la vanne à travers la pompe. Les deux premières phases par un polynôme d'ordre 1 et la dernière par un polynôme d'ordre 2.

D'où le Modèle qui suit :

Déclaration des variables:

- F Débit à travers la pompe
- N vitesse de la pompe

Le modèle pour le débit se décompose donc de la manière qui suit :

Dans la zone comprise entre 100mbar et 500mbar :

$$F = (-13.66 + 147.66 * P_{in} - 171.3287 * P^2 + 186.48 * P_{in}^3) * \frac{1}{3600 * 1000}$$

Dans la Zone comprise entre 1mbar et 3bar:

$$F = 474 - 285.05 * P_{in} + 105.43 * P_{in}^2 - 14.67 * P_{in}^3$$

Avec F en Nm³/s

P_{in} en bar

Afin de passer d'un débit exprimé en Nm³ à un débit en m³ dans le cas des gaz parfaits on utilise la loi des gaz parfaits pour chaque condition :

$$P_1 * V_1 = n * R * T_1 \quad \text{« Dans les conditions dites « Normales » »}$$

$$P_2 * V_2 = n * R * T_2 \quad \text{« Dans les conditions de fonctionnement »}$$

On dérive en fonction du temps, avec une hypothèse isotherme :

$$P_N * F_N = \dot{n} * R * T_N$$

$$P * F = \dot{n} * R * T$$

Le nombre de mol n étant égale et R une constante, la relation de conversion du débit de $\frac{Nm^3}{s}$ en $\frac{m^3}{s}$ s'exprime de la manière qui suit d'où l'expression qui suit :

$$F = F_N * \frac{P_N}{P_2} * \frac{T_2}{T_N}$$

Avec une équation finale respectant les unités de la bibliothèque

$$F = (-13.66 + 147.66 * P_{in} - 171.3287 * P^2 + 186.48 * P_{in}^3) * \frac{1}{3600 * 1000} * \frac{P_{atm}}{P_{in}} * \frac{T}{T_0}$$

Conservation des masses :

$$W_{out} = W_{In}$$

Conservation de chaque composant :

$$C_{in}^i = C_{Out}^i$$

Hypothèse isotherme :

$$T_{in} = T_{out}$$

k) BULLEUR

Le bulleur est un petit volume contenant un niveau mesuré avec précision d'huile à l'intérieur. Ce dispositif sert en cas de dépassant d'une pression seuil fixée par le niveau d'huile à évacuer le gaz en surpression, il est possible ajuster la pression maximale avant évacuation du gaz avec une précision relativement élevée pression dispose d'un port d'entrée de type gaz.

Le modèle calcule un débit à travers un orifice lorsqu'une valeur de pression seuil est atteinte :

Déclaration des variables:

- Safe_press pression de sécurité du bulleur
- D Diamètre d'entrée du gaz dans le bulleur

Équation :

Si la pression P_{in} est supérieure à Safe_press alors :

$$w_{in} = \frac{\pi * D^2 * \rho * 10^5}{128 * 10^{-3} * visc * ((p_{in} - Safe_{press}) * 10^5)}$$

Sinon :

$$w_{in} = 0$$

Publications liées à la thèse

Conférences

1. N. Bouchair, S. Charbonnier, P. Gayet. Fault isolation by comparison of event lists using a weighted distance, IEEE International Conference on Systems, Man and Cybernetics IEEE SMC 2013, Manchester, UK, 13-16 Octobre 2013
2. S. Charbonnier, N. Bouchair, P. Gayet. Assigning weights to alarms as a mean to learn information and assist supervision operators during alarms avalanches CM 2014 and MFPT 2014 The Eleventh International Conference on Condition Monitoring and Machinery Failure Prevention Technologies, 10-12 June 2014, Manchester, UK
3. S. Charbonnier, N. Bouchair, P. Gayet. Fault isolation by comparing alarm lists using a symbolic sequence matching algorithm submitted to the 19th IFAC World Congress, (IFAC WC 2014),Le Cap, South Africa, 2014
4. S. Charbonnier, N. Bouchair, P. Gayet. Analysis of fault diagnosability from SCADA alarms signatures using relevance indices, IEEE International Conference on Systems, Man and Cybernetics IEEE SMC 2014, San Diego, USA, 13-16 Octobre 2013

Articles

5. S. Charbonnier, N. Bouchair, P. Gayet A binary weighted distance to assist control operators during alarm flood situations" submitted to IEEE Transactions on Systems, Man and Cybernetics: Systems.

Sans cœur, nous ne sommes que des machines
Slogan publicitaire